listing circa ~1983
Donated by Charlie Kulas
5-15-18

```
                1000      .TITLE "UPN DOWN - for Atari systems"
                1020 ;
                1040 ; U.M65
                1060 ;
                1080 ;;   .OPT NO LIST
                1100      .SET 3,0
    =4000       1120 ORG =    $4000
0000            1140      *=  ORG
4000            1160 PSCOR0 *= *+1    ; player's score
4001            1180 PSCOR1 *= *+1
4002            1200 PSCOR2 *= *+1
4003            1220 ROUND *= *+1
4004            1240 SCRFLG2 *= *+1   ; flags VBI to scroll playfield
4005            1260 SNDCTL *= *+1    ; controls sound on/off
4006            1280 TRUCKX *= *+1
4007            1300 TRUCKY *= *+1
4008            1320 DSCROL *= *+1
4009            1340 SCRCTR *= *+1
400A            1360 SCRSPD *= *+1
400B            1380 SCROLF *= *+1
400C            1400 KX   *=  *+1
400D            1420 KY   *=  *+1
400E            1440 CX   *=  *+1
400F            1460 CY   *=  *+1
4010            1480 OLDVPOS *= *+1
4011            1500 OLDHPOS *= *+1
4012            1520 CHINDX *= *+1
4013            1540 CURCHAR *= *+1
4014            1560 SAVBUF *= *+40
403C            1580 DSPTBL *= *+223 ; table of address ptrs to screen (low)
411B            1600 DSPTBH *= *+223 ;          "                      (high)
41FA            1620 M4TBL *= *+16
420A            1640 CHARH *= *+64    ; addr. high for char index
424A            1660 CHARL *= *+64    ; addr. low for char index
428A            1680 ROWBUF *= *+10
4294            1700 CARLFT *= *+1
4295            1720 PLYRDAT *= *+16
42A5            1740 LINCNT *= *+1
42A6            1760 BUGX *= *+1
42A7            1780 BUGY *= *+1
42A8            1800 DLIST *= *+268
43B4            1820 PFDATA *= *+1024
    =0370       1840 HMEMSIZ = 880
47B4            1860 HMEM *= *+HMEMSIZ ; header mem
    =004D       1880 ATRACT = $4D
    =02FC       1900 CH   =   $02FC
    =02F2       1920 CH1  =   $02F2
4B24 000A141E   1940 M10TBL .BYTE 0,10,20,30,40,50
4B28 2832
4B2A 00050A0F   1960 M5TBL .BYTE 0,5,10,15,20,25,30,35
4B2E 14191E23
4B32 282D3237   1980      .BYTE 40,45,50,55,60,65,70,75
4B36 3C41464B
4B3A            2000 M3TBL *= *+80
```

```
4B8C 0000
4B8E 0000


4B90 0000
    =D40B       2060 VCOUNT = $D40B
    =D20A       2080 RANDOM = $D20A
    =0224       2100 VVBLKD = $0224
    =E462       2120 XITVBV = $E462
    =9000       2140 CHARS = $9000   ; start of char data (size= 4k)
    =7E70       2160 SMEM =  $8000-400 ; screen mem. (size=4480 bytes), must
                                       align to 4k
    =026C       2180 PFLEN = 620     ; # of Playfield chars
    =D01F       2200 CONSOL = $D01F
    =D40E       2220 NMIEN = $D40E
    =D40A       2240 WSYNC = $D40A
    =0230       2260 SDLSTL = $0230
    =0231       2280 SDLSTH = $0231
    =00D4       2300 R    =   $D4
    =00CA       2320 SCROLC = $CA
    =7000       2340 PMEM =  $7000   ; player/missile mem. (first 768 not used)
    =D000       2360 HPOSP0 = $D000
    =D001       2380 HPOSP1 = $D001
    =D002       2400 HPOSP2 = $D002
    =D003       2420 HPOSP3 = $D003
    =D004       2440 HPOSM0 = $D004
    =D005       2460 HPOSM1 = $D005
    =D006       2480 HPOSM2 = $D006
    =D007       2500 HPOSM3 = $D007
    =D407       2520 PMBASE = $D407
    =02C0       2540 PCOLR0 = $02C0
    =02C1       2560 PCOLR1 = $02C1
    =02C2       2580 PCOLR2 = $02C2
    =02C3       2600 PCOLR3 = $02C3
    =D012       2620 COLPM0 = $D012
    =D013       2640 COLPM1 = $D013
    =D014       2660 COLPM2 = $D014
    =D015       2680 COLPM3 = $D015
    =026F       2700 GPRIOR = $026F
    =D008       2720 SIZEP0 = $D008
    =D009       2740 SIZEP1 = $D009
    =D00A       2760 SIZEP2 = $D00A
    =D00B       2780 SIZEP3 = $D00B
    =E000       2800 OSCHARS = $E000
    =02F4       2820 CHBAS = $02F4
    =7300       2840 M0   =   PMEM+768
    =7400       2860 P0   =   PMEM+1024
    =7500       2880 P1   =   PMEM+1024+256
    =7600       2900 P2   =   PMEM+1024+512
    =7700       2920 P3   =   PMEM+1024+768
    =022F       2940 SDMCTL = $022F
    =D01D       2960 GRACTL = $D01D
    =7800       2980 IPF =    $7800   ; start of playfield data
    =D405       3000 VSCROL = $D405
    =0012       3020 RTCLOK = $12
                3040 ;
                3060 ;
4B92            3080      *=  ORG+$1000
5000 A200       3100      LDX #0       ; init stack
5002 9A         3120      TXS
```

```
5003 209453      3140      JSR IVARS
5006 200153      3160      JSR ITABLES


PAGE 3   UPN DOWN - for Atari systems                    D2:U.M65


5009 20C750      3180      JSR IPFDATA
500C 206B52      3200      JSR IDLIST
500F 20B255      3220      JSR ITCHARS
5012 20CC54      3240      JSR IHMEM
5015 200C52      3260      JSR ISMEM
5018 20FC50      3280      JSR IVBI
501B 208950      3300      JSR IPMEM
501E 201F64      3320      JSR IOBJTB  ; init OBJTBL
                 3340 ;;   JSR TINIT
5021 207267      3360      JSR ISFX    ; init sound effects
5024 20146E      3380      JSR IHFLGS
5027 A900        3400      LDA #0
5029 8DC569      3420      STA TYPREF
502C A901        3440      LDA #1
502E 8DC669      3460      STA TYPREF+1
5031 201269      3480      JSR DATEXP  ; expand data
5034 A900        3500      LDA #0
5036 8D0040      3520      STA PSCOR0  ; init player's score
5039 8D0140      3540      STA PSCOR1
503C 8D0240      3560      STA PSCOR2
503F 8D0340      3580      STA ROUND
5042 20BB65      3600      JSR NEWRND  ; setup next round
5045 A901        3620      LDA #1      ; sound on
5047 8D0540      3640      STA SNDCTL
504A A240        3660      LDX #64     ; set car's X
504C 8EA642      3680      STX BUGX
504F A946        3700      LDA #70
5051 8D155B      3720      STA OLDBUGY
5054 A030        3740      LDY #48     ; set car's Y
5056 8CA742      3760      STY BUGY
5059 A900        3780      LDA #0
505B 8D115B      3800      STA BUGDIR
505E A907        3820      LDA #7
5060 8D165B      3840      STA OLDBDIR
5063 20ED5E      3860      JSR SHOCAR
5066 A905        3880      LDA #5
5068 8D9442      3900      STA CARLFT
506B 208054      3920      JSR ECAR
                 3940 ;
506E A900        3960      LDA #0
5070 8D0A40      3980      STA SCRSPD
5073 8D1240      4000      STA CHINDX
                 4020 ;
5076 A93C        4040      LDA #$3C    ; enable joystick portA
5078 8D02D3      4060      STA PACTL
507B A980        4080      LDA #128
507D 8D185B      4100      STA JMPNDX
5080 A900        4120      LDA #0
5082 8D125B      4140      STA BPTHCT
5085 4C2151      4160      JMP MAIN10
                 4180 ;
5088             4200 ILOOP
5088 00          4220      BRK
                 4240 ;
                 4260 ;
                 4280 ;
5089             4300 IPMEM
5089 A970        4320      LDA # >PMEM ; init PMBASE
```

```
5089            4300 IPMEM
5089 A970       4320      LDA # >PMEM ; init PMBASE
```



```
508B 8D07D4     4340      STA PMBASE
508E A000       4360      LDY #0       ; clear PMEM
5090 A205       4380      LDX #5       ; 5*256 bytes
5092 A900       4400      LDA # <APMEM+768Ü ; set pointer to PMEM active area
5094 85D4       4420      STA R+0
5096 A973       4440      LDA # >APMEM+768Ü
5098 85D5       4460      STA R+1
509A A900       4480      LDA #0
509C            4500 IPME10
509C 91D4       4520      STA (R+0),Y
509E C8         4540      INY
509F D0FB       4560      BNE IPME10
50A1 CA         4580      DEX
50A2 F005       4600      BEQ IPME20
50A4 E6D5       4620      INC R+1
50A6 4C9C50     4640      JMP IPME10
50A9            4660 IPME20
50A9 A931       4680      LDA #$31     ; enable 3rd col. combo, missiles as 5th
                                       plyr
50AB 8D6F02     4700      STA GPRIOR
50AE A900       4720      LDA #0       ; 1x size for all
50B0 8D08D0     4740      STA SIZEP0
50B3 8D09D0     4760      STA SIZEP1
50B6 8D0AD0     4780      STA SIZEP2
50B9 8D0BD0     4800      STA SIZEP3
50BC A93E       4820      LDA #$3E
50BE 8D2F02     4840      STA SDMCTL
50C1 A903       4860      LDA #3
50C3 8D1DD0     4880      STA GRACTL
50C6 60         4900      RTS
                4920 ;
                4940 ;
50C7            4960 IPFDATA
50C7 A96C       4980      LDA # <PFLEN ; set # bytes to write
50C9 85D4       5000      STA R+0
50CB A902       5020      LDA # >PFLEN
50CD 85D5       5040      STA R+1
50CF A9B4       5060      LDA # <PFDATA ; set dest. ptr
50D1 85D6       5080      STA R+2
50D3 A943       5100      LDA # >PFDATA
50D5 85D7       5120      STA R+3
50D7 A900       5140      LDA # <IPF   ; set source ptr
50D9 85D8       5160      STA R+4
50DB A978       5180      LDA # >IPF
50DD 85D9       5200      STA R+5
50DF A000       5220      LDY #0
50E1            5240 IPFD10
50E1 B1D8       5260      LDA (R+4),Y
50E3 91D6       5280      STA (R+2),Y
50E5 C8         5300      INY
50E6 C000       5320      CPY #0
50E8 D0F7       5340      BNE IPFD10
50EA E6D9       5360      INC R+5
50EC E6D7       5380      INC R+3
50EE C6D5       5400      DEC R+1
50F0 10EF       5420      BPL IPFD10
50F2            5440 IPFD20
```

Moore Business Forms, Inc. sv 1412RH

```
50F4 91D6       5480     STA (R+2),Y
50F6 C8         5500     INY
50F7 C4D4       5520     CPY R+0
50F9 D0F7       5540     BNE IPFD20
50FB            5560 IPFD30
50FB 60         5580     RTS
                5600 ;
                5620 ;
                5640 ;
                5660 ;
50FC            5680 IVBI
50FC AD0ED4     5700     LDA NMIEN
50FF 293F       5720     AND #$3F    ; disable DLI, VBI
5101 8D0ED4     5740     STA NMIEN
5104 A9FA       5760     LDA # <DLI1
5106 8D0002     5780     STA $0200
5109 A953       5800     LDA # >DLI1
510B 8D0102     5820     STA $0201
510E A98E       5840     LDA # <VBINT
5110 8D2402     5860     STA VVBLKD
5113 A951       5880     LDA # >VBINT
5115 8D2502     5900     STA VVBLKD+1
5118 AD0ED4     5920     LDA NMIEN
511B 0940       5940     ORA #$40
511D 8D0ED4     5960     STA NMIEN
5120 60         5980     RTS
                6000 ;
                6020 ;
                6040 ;
                6060 ;
                6080 ;
5121            6100 MAIN10
5121 A900       6120     LDA #0      ; defeat attract
5123 854D       6140     STA ATRACT
5125 209367     6160     JSR SCROLL
5128 20CF5C     6180     JSR PMOVE
512B 20586E     6200     JSR FLGSCR
512E 208063     6220     JSR MOVE
5131 206264     6240     JSR TINIT
5134 20226D     6260     JSR FLGHIT
                6280 ;
                6300 ; Check for trucks hitting player
                6320 ;
5137 AD185B     6340     LDA JMPNDX  ; only do this if not jumping
513A 1022       6360     BPL MAIN15
513C 208965     6380     JSR COLISN
513F 901D       6400     BCC MAIN15
5141 A9FF       6420     LDA #-1     ; cancel truck
5143 9D1D63     6440     STA TYPE,X
5146 200E5A     6460     JSR ERASE   ; erase truck
5149 20EC65     6480     JSR VWDIE
514C CE9442     6500     DEC CARLFT
514F AD9442     6520     LDA CARLFT
5152 C906       6540     CMP #6
5154 9005       6560     BCC MAIN08
5156 A905       6580     LDA #5
5158 8D9442     6600     STA CARLFT
515B            6620 MAIN08
```

```
515B 208054     6640      JSR ECAR    ; show extra cars
                6660 ;
515E            6680 MAIN15
515E AD1FD0     6700      LDA CONSOL
5161 2902       6720      AND #2
5163 D008       6740      BNE MAIN20
5165 A900       6760      LDA #0
5167 8D0A40     6780      STA SCRSPD
516A 4C8851     6800      JMP MAIN50
516D            6820 MAIN20
516D AD1FD0     6840      LDA CONSOL
5170 2901       6860      AND #1
5172 D008       6880      BNE MAIN30
5174 A9FF       6900      LDA #-1
5176 8D0A40     6920      STA SCRSPD
5179 4C8851     6940      JMP MAIN50
517C            6960 MAIN30
517C AD1FD0     6980      LDA CONSOL
517F 2904       7000      AND #4
5181 D005       7020      BNE MAIN50
5183 A901       7040      LDA #1
5185 8D0A40     7060      STA SCRSPD
                7080 ;
5188            7100 MAIN50
5188 20EE5A     7120      JSR PAUSE
518B 4C2151     7140      JMP MAIN10
                7160 ;
                7180 ;
518E            7200 VBINT
                7220 ;
518E A21F       7240      LDX #31
5190            7260 VBIN05
5190 B5D4       7280      LDA R+0,X   ; save scratchpad regs.
5192 9D1440     7300      STA SAVBUF,X
5195 CA         7320      DEX
5196 10F8       7340      BPL VBIN05
                7360 ;
                7380 ; Set colors for header
                7400 ;
5198 A90A       7420      LDA #$0A    ; lum
519A 8DC502     7440      STA $02C5
519D A910       7460      LDA #$10    ; hue
519F 8DC602     7480      STA $02C6
51A2 A9FA       7500      LDA # <DLI1 ; set dli vector
51A4 8D0002     7520      STA $0200
51A7 A953       7540      LDA # >DLI1
51A9 8D0102     7560      STA $0201
                7580 ;
                7600 ; Setup header P/M flags
                7620 ;
51AC AC536E     7640      LDY HFPOS   ; HPOS
51AF 8C07D0     7660      STY HPOSM3
51B2 C8         7680      INY
51B3 C8         7700      INY
51B4 8C06D0     7720      STY HPOSM2
51B7 C8         7740      INY
51B8 C8         7760      INY
51B9 8C05D0     7780      STY HPOSM1
```

D2:U.M65

```
51BC C8          7800      INY
51BD C8          7820      INY
51BE 8C04D0      7840      STY HPOSM0
51C1 AD546E      7860      LDA HFPOS+1
51C4 8D00D0      7880      STA HPOSP0
51C7 AD556E      7900      LDA HFPOS+2
51CA 8D01D0      7920      STA HPOSP1
51CD AD566E      7940      LDA HFPOS+3
51D0 8D02D0      7960      STA HPOSP2
51D3 AD576E      7980      LDA HFPOS+4
51D6 8D03D0      8000      STA HPOSP3
                 8020 ;
51D9 ADE26D      8040      LDA FCOLOR  ; colors
51DC 8D19D0      8060      STA $D019
51DF ADE36D      8080      LDA FCOLOR+1
51E2 8D12D0      8100      STA COLPM0
51E5 ADE46D      8120      LDA FCOLOR+2
51E8 8D13D0      8140      STA COLPM1
51EB ADE56D      8160      LDA FCOLOR+3
51EE 8D14D0      8180      STA COLPM2
51F1 ADE66D      8200      LDA FCOLOR+4
51F4 8D15D0      8220      STA COLPM3
                 8240 ;
                 8260 ;
                 8280 ; Sounds
                 8300 ;
51F7 AD0540      8320      LDA SNDCTL  ; sound enabled?
51FA F003        8340      BEQ VBIN32  ; no - branch
51FC 20DD66      8360      JSR SFX     ; else do sounds
                 8380 ;
                 8400 ; Restore scratchpad regs
                 8420 ;
51FF             8440 VBIN32
51FF A21F        8460      LDX #31
5201             8480 VBIN40
5201 BD1440      8500      LDA SAVBUF,X
5204 95D4        8520      STA R+0,X
5206 CA          8540      DEX
5207 10F8        8560      BPL VBIN40
5209 4C62E4      8580      JMP XITVBV
                 8600 ;
                 8620 ; Init. the screen memory with 120 lines of PF
                 8640 ; pointed to by SCROL ptrs
                 8660 ;
520C             8680 ISMEM
                 8700 ; LDA NMIEN
                 8720 ; AND #$BF    ; disable VBI
                 8740 ; STA NMIEN
520C AD0B40      8760      LDA SCROLF
520F 48          8780      PHA
5210 A5CA        8800      LDA SCROLC
5212 48          8820      PHA
5213 A5CB        8840      LDA SCROLC+1
5215 48          8860      PHA
5216 A970        8880      LDA #112    ; do 112 PF lines
5218 85DD        8900      STA R+9
521A AE0840      8920      LDX DSCROL  ; get pointer to SMEM
521D             8940 ISME10
```

```
521D BD3C40     8960     LDA DSPTBL,X ; set (0,1) to SMEM begin
5220 85D4       8980     STA R+0
5222 BD1B41     9000     LDA DSPTBH,X
5225 85D5       9020     STA R+1
5227 20C868     9040     JSR PFLINE  ; write a PF line
                9060 ;
522A C6DD       9080     DEC R+9
522C F032       9100     BEQ ISME30
                9120 ;
522E E8         9140     INX          ; update screen pointer
522F EE0B40     9160     INC SCROLF
5232 AD0B40     9180     LDA SCROLF
5235 C910       9200     CMP #16
5237 D0E4       9220     BNE ISME10
5239 A900       9240     LDA #0
523B 8D0B40     9260     STA SCROLF
523E A5CA       9280     LDA SCROLC
5240 18         9300     CLC
5241 690A       9320     ADC #10
5243 85CA       9340     STA SCROLC
5245 A5CB       9360     LDA SCROLC+1
5247 6900       9380     ADC #0
5249 85CB       9400     STA SCROLC+1
524B C946       9420     CMP # >PFDATA+PFLEN
524D D0CE       9440     BNE ISME10
524F A5CA       9460     LDA SCROLC
5251 C920       9480     CMP # <PFDATA+PFLEN
5253 D0C8       9500     BNE ISME10
5255 A9B4       9520     LDA # <PFDATA ; reset
5257 85CA       9540     STA SCROLC
5259 A943       9560     LDA # >PFDATA
525B 85CB       9580     STA SCROLC+1
525D 4C1D52     9600     JMP ISME10
5260            9620 ISME30
5260 68         9640      PLA
5261 85CB       9660      STA SCROLC+1
5263 68         9680      PLA
5264 85CA       9700      STA SCROLC
5266 68         9720      PLA
5267 8D0B40     9740      STA SCROLF
                9760 ;
                9780 ; LDA NMIEN
                9800 ; ORA #$40
                9820 ; STA NMIEN
526A 60         9840      RTS
                9860 ;
                9880 ; Init. the DLIST
                9900 ;
526B            9920 IDLIST
526B A970       9940      LDA #$70
526D 8DA842     9960      STA DLIST
5270 8DA942     9980      STA DLIST+1
5273 8DAA42     010000    STA DLIST+2
5276 A942       010020    LDA #$42     ; 1 line mode 2+load
5278 8DAB42     010040    STA DLIST+3
527B A9B4       010060    LDA # <HMEM
527D 8DAC42     010080    STA DLIST+4
5280 A947       010100    LDA # >HMEM
```


Moore Business Forms, Inc. sv 14128H

```
5282 8DAD42     010120   STA DLIST+5
5285 A982       010140   LDA #$82    ; 1 line mode 2 + DLI
5287 8DAE42     010160   STA DLIST+6
528A A900       010180   LDA #$00    ; blank 1 scan line
528C 8DAF42     010200   STA DLIST+7
528F A90E       010220   LDA #$0E    ; 15 lines mode E
5291 A200       010240   LDX #0
5293            010260 IDLI10
5293 9DB042     010280   STA DLIST+8,X
5296 E8         010300   INX
5297 E00F       010320   CPX #15
5299 D0F8       010340   BNE IDLI10
529B A98E       010360   LDA #$8E    ; 1 line mode E + DLI (top flag DLI)
529D 8DBF42     010380   STA DLIST+23
                010400 ;
52A0 A98D       010420   LDA #$8D    ; 1 line mode D + DLI (blank line to
                                      separate header and playfield)
52A2 8DC042     010440   STA DLIST+24
                010460 ;
52A5 A200       010480   LDX #0
52A7 A96D       010500   LDA #$6D
52A9            010520 IDLI30
52A9 9DC142     010540   STA DLIST+25,X ; 80 lines mode D w/load + vert. scroll
                                          enable
52AC E8         010560   INX
52AD E8         010580   INX
52AE E8         010600   INX
52AF E0F0       010620   CPX #240
52B1 D0F6       010640   BNE IDLI30
                010660 ;
52B3 8DAE43     010680   STA DLIST+262
52B6 A941       010700   LDA #$41    ; jump and wait instr.
52B8 8DB143     010720   STA DLIST+265
52BB A9A8       010740   LDA # <DLIST
52BD 8DB243     010760   STA DLIST+266
52C0 A942       010780   LDA # >DLIST
52C2 8DB343     010800   STA DLIST+267
52C5 A900       010820   LDA #0      ; reset pointer for DLIST scroll
52C7 8D0840     010840   STA DSCROL  ; to top of SMEM
52CA 8D0B40     010860   STA SCROLF
52CD A9B4       010880   LDA # <PFDATA
52CF 85CA       010900   STA SCROLC
52D1 A943       010920   LDA # >PFDATA
52D3 85CB       010940   STA SCROLC+1
52D5 20E352     010960   JSR CHGDSP  ; set DLIST pointers
52D8 A9A8       010980   LDA # <DLIST ; select DLIST
52DA 8D3002     011000   STA SDLSTL
52DD A942       011020   LDA # >DLIST
52DF 8D3102     011040   STA SDLSTH
52E2 60         011060   RTS
                011080 ;
                011100 ; Write the display list pointers
                011120 ; using the value in DSCROL to
                011140 ; determine tne first line.
52E3            011160 CHGDSP
52E3 AD0840     011180   LDA DSCROL  ; use to index first line
52E6 18         011200   CLC         ; first line is +16 to leave room for
```

```
52E7 6910       011220   ADC #16
52E9 AA         011240   TAX
52EA A000       011260   LDY #0
52EC            011280 MOVD10
52EC BD3C40     011300   LDA DSPTBL,X ; get low byte of pointer
52EF 99C242     011320   STA DLIST+26,Y ; write to DLIST
52F2 BD1B41     011340   LDA DSPTBH,X ;     high byte
52F5 99C342     011360   STA DLIST+27,Y
52F8 E8         011380   INX          ; next DSPTBL value
52F9 C8         011400   INY          ; next DLIST pointer
52FA C8         011420   INY
52FB C8         011440   INY
52FC C0F0       011460   CPY #240     ; 80 lines
52FE D0EC       011480   BNE MOVD10
5300 60         011500   RTS
                011520 ; INIT THE TABLES
                011540 ;
                011560 ; Setup DSPTBL,DSPTBH with pointer
                011580 ; values for the display list
                011600 ; scroll.  Though there are 80
                011620 ; There are 223 entries in DSPTB,  112 to
                011640 ; handle the addresses of the 80 displayed mode
                011660 ; lines, 16 for "before" and 16 for "after" buffers,
                011680 ; 111 to extend the table so that even
                011700 ; when indexing starting at line #111
                011720 ; there will be no need to wraparound the index
                011740 ;
                011760 ;
5301            011780 ITABLES
5301 A270       011800   LDX #112    ; do first 80 pointers
5303 A93C       011820   LDA # <DSPTBL ; set pointer to DSPTBL
5305 85D6       011840   STA R+2
5307 A940       011860   LDA # >DSPTBL
5309 85D7       011880   STA R+3
530B A000       011900   LDY #0       ; index to DSPTBL
530D            011920 ITAB05
530D A970       011940   LDA # <SMEM ; set initial pointer value
530F 85D4       011960   STA R+0
5311 A97E       011980   LDA # >SMEM
5313 85D5       012000   STA R+1
5315            012020 ITAB10
5315 A5D5       012040   LDA R+1      ; store a pointer
5317 991B41     012060   STA DSPTBH,Y ; high byte
531A A5D4       012080   LDA R+0
531C 993C40     012100   STA DSPTBL,Y ; low byte
531F C8         012120   INY
5320 CA         012140   DEX
5321 F01D       012160   BEQ ITAB20
                012180 ;
                012200 ; LDA R+1      ; save high nybble for 4k boundary check
                012220 ; AND #$F0
                012240 ; STA R+4
                012260 ; LDA R+0
5323 18         012280   CLC          ; update pointer value
5324 6928       012300   ADC #40      ; 40 bytes/mode line
5326 85D4       012320   STA R+0
5328 90EB       012340   BCC ITAB10
532A E6D5       012360   INC R+1
```



```
532C 4C1553     012380   JMP ITAB10  ; next pointer
                012400 ;
```

```
5328 90EB       012340   BCC ITAB10
532A E6D5       012360   INC R+1
```



```
532C 4C1553     012380   JMP ITAB10  ; next pointer
                012400 ;
                012420 ; Check for 4k boundary crossing
                012440 ;
532F            012460 ITAB15
532F A5D5       012480   LDA R+1
5331 29F0       012500   AND #$F0
5333 C5D8       012520   CMP R+4
5335 F0DE       012540   BEQ ITAB10  ; ok - branch
5337 85D5       012560   STA R+1     ; else force to 4k
5339 A900       012580   LDA #$00
533B 85D4       012600   STA R+0
533D 4C1553     012620   JMP ITAB10
                012640 ;
                012660 ; do last 111 pointer values
                012680 ;
5340            012700 ITAB20
5340 A5D6       012720   LDA R+2     ; is this the first time here?
5342 C93C       012740   CMP # <DSPTBL
5344 D00E       012760   BNE ITAB30  ; no - already did this, branch
5346 18         012780   CLC         ; move pointer to DSPTBL
5347 6970       012800   ADC #112
5349 85D6       012820   STA R+2
534B 9002       012840   BCC ITAB28
534D E6D7       012860   INC R+3
534F            012880 ITAB28
534F A26F       012900   LDX #111    ; 111 more values
5351 4C0D53     012920   JMP ITAB05
                012940   ;
                012960 ; init CHARH,CHARL index to char
                012980 ; addresses.
                013000 ;
5354            013020 ITAB30
5354 A900       013040   LDA # <CHARS ; init first addr to CHARS start
5356 85D4       013060   STA R+0
5358 A990       013080   LDA # >CHARS
535A 85D5       013100   STA R+1
535C A000       013120   LDY #0
535E            013140 ITAB40
535E A5D5       013160   LDA R+1     ; write a pointer
5360 990A42     013180   STA CHARH,Y
5363 A5D4       013200   LDA R+0
5365 994A42     013220   STA CHARL,Y
                013240 ;
5368 C8         013260   INY
5369 C040       013280   CPY #64     ; do for # chars in CHARS
536B F00C       013300   BEQ ITAB50
                013320   ;
536D 18         013340   CLC
536E 6940       013360   ADC #64
5370 85D4       013380   STA R+0
5372 90EA       013400   BCC ITAB40
5374 E6D5       013420   INC R+1
5376 4C5E53     013440   JMP ITAB40
5379            013460 ITAB50
                013480 ;
                013500 ; init mult. by 4 table for fine
                013520 ; scroll add in calc.
```

```
                    013500 ; init mult. by 4 table for fine
                    013520 ; scroll add in calc.
```



```
                    013540 ;
5379 A93C           013560   LDA #60
537B A00F           013580   LDY #15
537D 38             013600   SEC          ; set for subt.
537E                013620 ITAB60
537E 99FA41         013640   STA M4TBL,Y
5381 E904           013660   SBC #4
5383 88             013680   DEY
5384 10F8           013700   BPL ITAB60
                    013720 ;
                    013740 ; init *3 table
                    013760 ;
5386 A9F0           013780   LDA #80*3    ; 80 entries
5388 A24F           013800   LDX #79
538A 38             013820   SEC
538B                013840 ITAB70
538B E903           013860   SBC #3
538D 9D3A4B         013880   STA M3TBL,X
5390 CA             013900   DEX
5391 10F8           013920   BPL ITAB70
                    013940 ;
5393 60             013960   RTS
                    013980 ;
                    014000 ; INIT THE VARIABLES
                    014020 ;
5394                014040 IVARS
5394 A900           014060   LDA #0
5396 8DFB6F         014080   STA FTOPDLI
                    014100 ;
5399 A950           014120   LDA #FDLY    ; init flag delay counter
539B 8D216D         014140   STA FDLYCT
539E A900           014160   LDA #0       ; no flag toggle
53A0 8D1F6D         014180   STA FTOGL
                    014200 ;
53A3 A200           014220   LDX #OBJMAX-1
53A5 A9FF           014240   LDA #$FF
53A7                014260 IVAR10
53A7 9D7763         014280   STA OVRLAY,X
53AA CA             014300   DEX
53AB 10FA           014320   BPL IVAR10
                    014340 ;
53AD 60             014360   RTS
                    014380 ;
                    014400 ;
                    014420   .INCLUDE #D2:HDRSUB.M65
                    1000     .PAGE "header subroutines"
```

```
                    1020 ; HDRSUB.M65
                    1040 ; SUBROUTINES TO HANDLE THE
                    1060 ; HEADER DISPLAY FOR UPN DOWN
                    1080 ;
                    1100 ;
                    1120 ;
                    1140 ;
                    1160 ; *ADDSCOR*
                    1180 ; ADD THE 2 BYTE BCD VALUE IN R+5,R+6
                    1200 ; TO THE 3 BYTE BCD VALUE IN R+2,R+3,R+4
                    1220 ; AND DISPLAY AS A 6-DIGIT DECIMAL
                    1240 ; NUMBER AT SCREEN LOCATION POINTED
                    1260 ; TO BY R+0,R+1
                    1280 ;
53AE                1300 ADDSCOR
53AE F8             1320     SED          ; decimal mode
53AF A5D6           1340     LDA R+2      ; get LSB
53B1 18             1360     CLC
53B2 65D9           1380     ADC R+5
53B4 85D6           1400     STA R+2
53B6 A5D7           1420     LDA R+3      ; middle byte
53B8 65DA           1440     ADC R+6
53BA 85D7           1460     STA R+3
53BC A5D8           1480     LDA R+4
53BE 6900           1500     ADC #0
53C0 85D8           1520     STA R+4
53C2                1540 ADDS10
53C2 A005           1560     LDY #5       ; ptr to dest.
53C4 A200           1580     LDX #0       ; index to byte being written
53C6                1600 ADDS20
53C6 B5D6           1620     LDA R+2,X    ; get byte
53C8 290F           1640     AND #$0F     ; write low nybble
53CA 91D4           1660     STA (R+0),Y
53CC B5D6           1680     LDA R+2,X    ; write high nybble
53CE 4A             1700     LSR A
53CF 4A             1720     LSR A
53D0 4A             1740     LSR A
53D1 4A             1760     LSR A
53D2 88             1780     DEY
53D3 91D4           1800     STA (R+0),Y
53D5 E8             1820     INX
53D6 88             1840     DEY
53D7 10ED           1860     BPL ADDS20
53D9 D8             1880     CLD
53DA 60             1900     RTS
                    1920 ;
                    1940 ;
                    1960 ;
                    1980 ; INIT THE PLAYER DATA TABLE
                    2000 ;
53DB                2020 IPLYRDT
53DB A900           2040     LDA #0
53DD A20F           2060     LDX #15
53DF                2080 IPLY10
53DF 9D9542         2100     STA PLYRDAT,X ; clear PLYRDAT data area
53E2 CA             2120     DEX
53E3 10FA           2140     BPL IPLY10
                    2160 ;
```

```
53E3 10FA       2140      BPL IPLY10
                2160 ;
```



```
53E5 A9BA       2180      LDA # <HMEM+6 ; set pointer to PLYR1 score disp
53E7 8D9542     2200      STA PLYRDAT
53EA A947       2220      LDA # >HMEM+6
53EC 8D9642     2240      STA PLYRDAT+1
53EF A9E2       2260      LDA # <HMEM+46 ;      "          PLYR2      "
53F1 8D9D42     2280      STA PLYRDAT+8
53F4 A947       2300      LDA # >HMEM+46
53F6 8D9D42     2320      STA PLYRDAT+8
53F9 60         2340      RTS
                2360 ;
                2380 ; DLI routine to change colors
                2400 ; between header text (top) and
                2420 ; header graphics (bottom) areas
                2440 ;
53FA            2460 DLI1
53FA 48         2480      PHA
53FB A900       2500      LDA #$00    ; black
53FD 8D0AD4     2520      STA WSYNC
5400 8D1AD0     2540      STA $D01A
5403 A90E       2560      LDA #$0E    ; white
5405 8D16D0     2580      STA $D016
5408 A986       2600      LDA #$86    ; blue
540A 8D17D0     2620      STA $D017
540D A91E       2640      LDA #$1E    ; yellow
540F 8D18D0     2660      STA $D018
5412 A965       2680      LDA # <TFDLI ; set vector to next DLI
5414 8D0002     2700      STA $0200
5417 A954       2720      LDA # >TFDLI
5419 8D0102     2740      STA $0201
541C 68         2760      PLA
541D 40         2780      RTI
                2800 ;
                2820 ;
541E            2840 DLI2
541E 48         2860      PHA
541F A9C6       2880      LDA #$C6    ; green
5421 8D0AD4     2900      STA WSYNC
5424 8D1AD0     2920      STA $D01A
5427 A900       2940      LDA #0      ;black
5429 8D18D0     2960      STA $D018
542C A9F6       2980      LDA #$F6    ;brown
542E 8D16D0     3000      STA $D016
5431 A9B9       3020      LDA # <FDLI ; set flag DLI vect.
5433 8D0002     3040      STA $0200
5436 A96F       3060      LDA # >FDLI
5438 8D0102     3080      STA $0201
                3100 ;
                3102 ; Restore VW colors
                3104 ;
543B A99A       3120      LDA #$9A    ; blue
543D 8D12D0     3140      STA COLPM0
5440 8D14D0     3160      STA COLPM2
5443 A906       3180      LDA #$06    ; grey
5445 8D13D0     3200      STA COLPM1
5448 8D15D0     3220      STA COLPM3
                3240 ;
                3241 ; Restore VW position
                3242 ;
```

```
                        3241 ; Restore VW position
                        3242 ;
```



```
5448 AD1860     3243    LDA BUGH0
544B 8D00D0     3244    STA HPOSP0
544E AD1960     3245    LDA BUGH1
5451 8D01D0     3246    STA HPOSP1
5454 AD1A60     3247    LDA BUGH2
5457 8D02D0     3248    STA HPOSP2
545A AD1B60     3249    LDA BUGH3
545D 8D03D0     3250    STA HPOSP3
                3251 ;
5460 68         3260    PLA
5461 40         3280    RTI
                3300 ;
                3320 ; TOP FLAG DLI
                3340 ;
5462            3360 TFDLI
5462 48         3380    PHA
5463 A91E       3400    LDA # <DLI2
5465 8D0002     3420    STA $0200
5468 A954       3440    LDA # >DLI2
546A 8D0102     3460    STA $0201
546D A900       3480    LDA #0      ; set index to FDLISEQ
546F 8DFA6F     3500    STA FDLINDX
5472 ADFB6F     3520    LDA FTOPDLI ; need top DLI?
5475 F004       3540    BEQ DLI220
5477 68         3560    PLA         ; yes restore stack
5478 4CB96F     3580    JMP FDLI    ; do flag DLI
547B            3600 DLI220
547B 68         3620    PLA
547C 40         3640    RTI
                3660 ;
                3680 ;
                3700 ; DRAW THE PLAYERS EXTRA CARS
                3720 ; IN HMEM.  NUMBER OF CARS DRAWN
                3740 ; IS "CARLFT"
                3760 ;
547D            3780 ECAR
547D A954       3800    LDA # <HMEM+160 ; init ptr to HMEM
547F 85D4       3820    STA R+0
5481 A948       3840    LDA # >HMEM+160
5483 85D5       3860    STA R+1
5485 A200       3880    LDX #0
5487            3900 ECAR10
5487 A000       3920    LDY #0
5489 84DD       3940    STY R+9
548B            3960 ECAR20
548B A5DD       3980    LDA R+9     ; get count of chars drawn
548D C905       4000    CMP #5
548F F023       4020    BEQ ECAR50  ; done with row
5491 CD9442     4040    CMP CARLFT  ; check written enough cars yet
5494 B00F       4060    BCS ECAR30  ; yes - branch to write blanks
5496 BD4A55     4080    LDA ECARDAT,X ; write car data, get byte
5499 91D4       4100    STA (R+0),Y
549B E8         4120    INX         ; write next byte
549C C8         4140    INY
549D BD4A55     4160    LDA ECARDAT,X
54A0 91D4       4180    STA (R+0),Y
54A2 4CB054     4200    JMP ECAR40
54A5            4220 ECAR30
```

```
54A5 4CB054      4200      JMP ECAR40
54A8             4220 ECAR30
```



```
54A8 A900        4240      LDA #0        ; write blanks
54AA 91D4        4260      STA (R+0),Y
54AC E8          4280      INX
54AD C8          4300      INY
54AE 91D4        4320      STA (R+0),Y
54B0             4340 ECAR40
54B0 E6DD        4360      INC R+9       ; inc count of cars drawn on row
54B2 CA          4380      DEX           ; continue - set X back to data for next
                                         car
54B3 C8          4400      INY           ; next space
54B4 4C8E54      4420      JMP ECAR20    ; next character
                 4440 ;
54B7             4460 ECAR50
54B7 E016        4480      CPX #22       ; check done (12 rows*2 bytes/row  -1 for
                                         old index -1 for starting at 0
54B9 B010        4500      BCS ECAR60
54BB E8          4520      INX           ; set to new data
54BC E8          4540      INX
54BD A5D4        4560      LDA R+0       ; update HMEM ptr to next row
54BF 18          4580      CLC
54C0 6928        4600      ADC #40       ; 40 bytes/line
54C2 85D4        4620      STA R+0
54C4 90C4        4640      BCC ECAR10
54C6 E6D5        4660      INC R+1
54C8 4C8A54      4680      JMP ECAR10    ; do next row
54CB             4700 ECAR60
54CB 60          4720      RTS
                 4740 ;
                 4760 ; INIT THE HEADER MEM.
                 4780 ;
54CC             4800 IHMEM
54CC A24F        4820      LDX #79       ; init text from table
54CE             4840 IHME10
54CE BD6255      4860      LDA IHTXT,X
54D1 9DB447      4880      STA HMEM,X
54D4 CA          4900      DEX
54D5 10F7        4920      BPL IHME10
                 4940 ;
                 4960 ; clear next header bytes for mode E
                 4980 ;
54D7 A203        5000      LDX # >HMEMSIZ-80
54D9 A000        5020      LDY #0
54DB A904        5040      LDA # <HMEM+80
54DD 85D4        5060      STA R+0
54DF A948        5080      LDA # >HMEM+80
54E1 85D5        5100      STA R+1
54E3 A900        5120      LDA #0
54E5             5140 IHME20
54E5 91D4        5160      STA (R+0),Y
54E7 C8          5180      INY
54E8 D0FB        5200      BNE IHME20
54EA CA          5220      DEX
54EB F005        5240      BEQ IHME30
54ED E6D5        5260      INC R+1       ; next page
54EF 4CE554      5280      JMP IHME20
54F2             5300 IHME30
54F2 A220        5320      LDX # <HMEMSIZ-80
54F4             5340 IHME40
```

Moore Business Forms, Inc. sv 1412RH

```
54F2 A220          5320      LDX # <HMEMSIZ-80
54F4               5340 IHME40
```



```
54F4 91D4          5360      STA (R+0),Y
54F6 C8            5380      INY
54F7 CA            5400      DEX
54F8 D0FA          5420      BNE IHME40
54FA 60            5840      RTS
                   5860 ;
                   5880 ; DRAW A FLAG TO A MODE "D" OR
                   5900 ; A MODE "E" SCREEN.
                   5920 ;
                   5940 ; Dest. is in (R+0,R+1)
                   5960 ; Color mask is passed in R+3
                   5980 ;
                   6000 ; R+3=$00     black (invisible)
                   6020 ;     $55     white
                   6040 ;     $AA     blue
                   6060 ;     $FF     green
                   6220 ;
54FB               6280 EFLAG
54FB 8A            6282      TXA
54FC 48            6284      PHA
                   6380 ;
54FD A200          6400      LDX #0
54FF A000          6420      LDY #0
5501               6440 EFLA10
5501 BD2A55        6460      LDA FLAGDAT,X ; get data
5504 25D7          6470      AND R+3       ; set color
5506 91D4          6480      STA (R+0),Y
5508 E8            6540      INX           ; next data
5509 BD2A55        6560      LDA FLAGDAT,X
550C 25D7          6570      AND R+3
550E A001          6580      LDY #1        ; write 2nd column (figure is 2 bytes
                             wide)
5510 91D4          6600      STA (R+0),Y
5512 E8            6660      INX
5513 E020          6680      CPX #32       ; done?
5515 F010          6700      BEQ EFLA20    ; yes - branch
5517 A000          6720      LDY #0        ; else next line (inc by 2 lines)
5519 A5D4          6740      LDA R+0
551B 18            6760      CLC
551C 6928          6780      ADC #40
551E 85D4          6800      STA R+0
5520 90DF          6820      BCC EFLA10
5522 E6D5          6840      INC R+1
5524 4C0155        6860      JMP EFLA10
5527               6880 EFLA20
                   6900 ;
5527 68            6920      PLA
5528 AA            6940      TAX
5529 60            7000      RTS
                   7020 ;
                   7040 ; data for flag
                   7060 ;
552A               7080 FLAGDAT
                   7100      ;          FLAG DATA FOR MODE E
552A 0000          7120      .BYTE $00,$00
552C 3F00          7140      .BYTE $3F,$00
552E 3FC0          7160      .BYTE $3F,$C0
5530 3FF0          7180      .BYTE $3F,$F0
```

Moore Business Forms, Inc. sv - 1412RM

```
552E 3FC0          7160      .BYTE $3F,$C0
5530 3FF0          7180      .BYTE $3F,$F0
```



```
5532 3FFC          7200      .BYTE $3F,$FC
5534 3FFC          7220      .BYTE $3F,$FC
5536 3FF0          7240      .BYTE $3F,$F0
5538 3FC0          7260      .BYTE $3F,$C0
553A 3F00          7280      .BYTE $3F,$00
553C 3000          7300      .BYTE $30,$00
553E 3000          7320      .BYTE $30,$00
5540 3000          7340      .BYTE $30,$00
5542 3000          7360      .BYTE $30,$00
5544 3000          7380      .BYTE $30,$00
5546 3000          7400      .BYTE $30,$00
5548 0000          7420      .BYTE $00,$00
                   7440      ;          END OF FLAG DATA
                   7460 ;
                   7480 ; data for player's extra cars
                   7500 ; (12 rows x 2 columns)
554A               7520 ECARDAT
                   7540 ;        .BYTE 242,143,250,175,234,171
                   7560 ;    .BYTE 42,168,32,8,32,8,32,8,32,8,34,136,234,171,250,17
                                    5,240,15
554A 00000690      7580      .BYTE 0,0,6,144,58,172,58,172,57,108,5,80
554E 3AAC3AAC
5552 396C0550
5556 05500550      7600      .BYTE 5,80,5,80,58,172,58,172,62,188,60,60
555A 3AAC3AAC
555E 3EBC3C3C
                   7620 ;
                   7640 ;
                   7660 ; table to init HMEM with chars
                   7680 ;
5562               7700 IHTXT
5562 100F1312      7720      .BYTE 16,15,19,18,1,24,0,0,0,0,0,0,24
5566 01180000
556A 00000000
556E 18
556F 191A1B1C      7740      .BYTE 25,26,27,28,29,30,31,32,33,34,35,36,37,38
5573 1D1E1F20
5577 21222324
557B 2526
557D 180D0E0C      7760      .BYTE 24,13,14,12,13,24,24,0,0,0,0,0,0
5581 0D181800
5585 00000000
5589 00
558A 100F1312      7780      .BYTE 16,15,19,18,2,24,0,0,0,0,0,0,24
558E 02180000
5592 00000000
5596 18
5597 2728292A      7800      .BYTE 39,40,41,42,43,44,45,46,47,48,49,50,51,52
559B 2B2C2D2E
559F 2F303132
55A3 3334
55A5 18121415      7820      .BYTE 24,18,20,21,22,23,24,0,0,0,0,0,0
55A9 16171800
55AD 00000000
55B1 00
                   7840 ;
```

```
                    7900 ; OS char's needed for text are copied
                    7920 ; to TCHAR mem. and CHBASE is set
                    7940 ; to point to TCHAR
                    7960 ;
55B2                7980 ITCHARS
55B2 A900           8000     LDA # <TCHARS ; init dest. ptr. to TCHAR area
55B4 85D6           8020     STA R+2
55B6 A958           8040     LDA # >TCHARS
55B8 85D7           8060     STA R+3
55BA A200           8080     LDX #0       ; init index to char#'s to be copied
55BC                8100 ITCH10
55BC BDF555         8120     LDA CHCOPY,X ; get char# to copy
55BF 302E           8140     BMI ITCH30   ; 0 is end of data
55C1 A000           8160     LDY #0
55C3 84D5           8180     STY R+1
55C5 0A             8200     ASL A        ; 8 bytes/char
55C6 26D5           8220     ROL R+1
55C8 0A             8240     ASL A
55C9 26D5           8260     ROL R+1
55CB 0A             8280     ASL A
55CC 26D5           8300     ROL R+1
55CE 85D4           8320     STA R+0
55D0 A5D5           8340     LDA R+1      ; add in offset to OS char base
55D2 18             8360     CLC
55D3 69E0           8380     ADC # >OSCHARS
55D5 85D5           8400     STA R+1
                    8420 ;
55D7                8440 ITCH15
55D7 A007           8460     LDY #7
55D9                8480 ITCH20
55D9 B1D4           8500     LDA (R+0),Y
55DB 91D6           8520     STA (R+2),Y
55DD 88             8540     DEY
55DE 10F9           8560     BPL ITCH20
                    8580 ;
55E0 E8             8600     INX          ; next char
55E1 A5D6           8620     LDA R+2      ; inc dest ptr
55E3 18             8640     CLC
55E4 6908           8660     ADC #8
55E6 85D6           8680     STA R+2
55E8 90D2           8700     BCC ITCH10
55EA E6D7           8720     INC R+3
55EC 4CBC55         8740     JMP ITCH10
55EF                8760 ITCH30
55EF A958           8780     LDA # >TCHARS ; set CHBAS to new charset
55F1 8DF402         8800     STA CHBAS
55F4 60             8820     RTS
                    8840 ; CHCOPY is a list of char#'s to
                    8860 ; be copied from the OS charset
                    8880 ;
                    8900 ; chars copied are:
55F5                8920 CHCOPY
                    8940 ;              0, 1, 2, 3, 4, 5, 6, 7
55F5 10111213       8960     .BYTE 16,17,18,19,20,21,22,23
55F9 14151617
                    8980 ;              8, 9, A, E, G, H, I, L
55FD 18192125       9000     .BYTE 24,25,33,37,39,40,41,44
5601 2728292C
```

Moore Business Forms, Inc. 141ZRH

```
55FD 18192125   9000      .BYTE 24,25,33,37,39,40,41,44
5601 2728292C

PAGE 20   UPN DOWN - for Atari systems
header subroutines                                  D2:HDRSUB.M65

                9020 ;          P, Q, R, Y, O, U, N, D
5605 30313239   9040      .BYTE 48,49,50,57,47,53,46,36
5609 2F352E24
                9060 ;          space, end
560D 0080       9080      .BYTE 0,128
                9100 ;
                9120 ; Reserve space for TCHAR
                9140 ; (title and text) chars
                9160 ;    must align to 1K
560F            9180      .DS AA*+1023&$FC00-*
5800            9200 TCHARS
5800            9220      .DS 25*8    ; reserve for OS char copy
                9240 ; UPN DOWN title chars are at #25-52
58C8 0F1F1E1E   9260      .BYTE 15,31,30,30,62,60,124,120
58CC 3E3C7C78
58D0 0F0F0F1F   9280      .BYTE 15,15,15,31,30,30,60,61
58D4 1E1E3C3D
58D8 8080000E   9300      .BYTE 128,128,0,14,15,30,28,61
58DC 0F1E1C3D
58E0 000000E1   9320      .BYTE 0,0,0,225,243,115,231,231
58E4 F373E7E7
58E8 000000DF   9340      .BYTE 0,0,0,223,183,231,207,143
58EC B7E7CF8F
58F0 00000000   9360      .BYTE 0,0,0,0,128,128,0,0
58F4 80800000
58F8 0F1F3F00   9380      .BYTE 15,31,63,0,30,62,124,120
58FC 1E3E7C78
5900 FCFFFF07   9400      .BYTE 252,255,255,7,15,15,15,31
5904 0F0F0F1F
5908 00008083   9420      .BYTE 0,0,128,131,143,30,60,56
590C 8F1E3C38
5910 000000F0   9440      .BYTE 0,0,0,240,249,57,123,243
5914 F9397BF3
5918 000000F3   9460      .BYTE 0,0,0,243,227,199,207,159
591C E3C7CF9F
5920 000000C7   9480      .BYTE 0,0,0,199,199,143,30,28
5924 C78F1E1C
5928 0000008E   9500      .BYTE 0,0,0,142,157,31,62,60
592C 9D1F3E3C
5930 000000F8   9520      .BYTE 0,0,0,248,188,60,56,120
5934 BC3C3878
5938 783F3F00   9540      .BYTE 120,63,63,0,0,0,0,0
593C 00000000
5940 FFFE7806   9560      .BYTE 255,254,120,6,31,63,31,15
5944 1F3F1F0F
5948 397F77E0   9580      .BYTE 57,127,119,224,192,192,128,0
594C C0C08000
5950 CF8E9E00   9600      .BYTE 207,142,158,0,0,0,0,0
5954 00000000
5958 0E1E1C3D   9620      .BYTE 14,30,28,61,63,63,63,30
595C 3F3F3F1E
5960 00000180   9640      .BYTE 0,0,1,128,192,192,128,0
5964 C0C08000
5968 F8FFFF00   9660      .BYTE 248,255,255,0,0,0,0,0
596C 00000000
5970 7FFCF800   9680      .BYTE 127,252,248,0,0,0,0,0
5974 00000000
5978 3D1F0F00   9700      .BYTE 61,31,15,0,0,0,0,0
```

Moore Business Forms, Inc. sv 14128M

```
5964 C0C08000
5968 F8FFFF00   9660      .BYTE 248,255,255,0,0,0,0,0
596C 00000000
5970 7FFCF800   9680      .BYTE 127,252,248,0,0,0,0,0
5974 00000000
5978 3D1F0F00   9700      .BYTE 61,31,15,0,0,0,0,0
```



```
597C 00000000
5980 F7C78700   9720      .BYTE 247,199,135,0,0,0,0,0
5984 00000000
5988 9EFF1F00   9740      .BYTE 158,255,31,0,0,0,0,0
598C 00000000
5990 3CFCBE00   9760      .BYTE 60,252,190,0,0,0,0,0
5994 00000000
5998 7870F001   9780      .BYTE 120,112,240,1,1,1,1,0
599C 01010100
59A0 70F0E0EC   9800      .BYTE 112,240,224,236,254,254,252,240
59A4 FEFEFCF0
                014440    .INCLUDE #D2:DRAW.M65
                1000      .PAGE "DRAW.M65"
```

Moore Business Forms, Inc. 14128R

```
                        1020 ;
                        1040 ; DRAW ROUTINE - DRAWS OBJECT
                        1060 ; IN OBJTBL INDEXED BY Xreg
                        1080 ;
59A8                    1100 DRAW
59A8 8A                 1120     TXA          ; save OBJTBL index
59A9 48                 1140     PHA
                        1160 ;
59AA 202A64             1180     JSR OVRLAP
59AD C0FF               1200     CPY #$FF
59AF F012               1220     BEQ DRAW01
59B1                    1240 DRA110
59B1 B97763             1260     LDA OVRLAY,Y
59B4 C9FF               1280     CMP #$FF
59B6 F004               1300     BEQ DRA100
59B8 A8                 1320     TAY
59B9 4CB159             1340     JMP DRA110
59BC                    1360 DRA100
59BC 8A                 1380     TXA
59BD 997763             1400     STA OVRLAY,Y ; set overlapped obj ptr to this object
59C0 4CC759             1420     JMP DRAW02
59C3                    1440 DRAW01
59C3 98                 1450     TYA
59C4 9D7763             1460     STA OVRLAY,X ; this obj doesn't overlap anything
59C7                    1480 DRAW02
59C7 A901               1500     LDA #1       ; flag draw setup
59C9 20695A             1520     JSR DESETUP ; setup pointers and variables for draw
59CC B03D               1540     BCS DRAW30  ; branch if not displayable
                        1560 ;
                        1580 ;
59CE                    1600 DRAW04
59CE A910               1620     LDA #16      ; # rows to draw
59D0 85E7               1640     STA R+19
59D2 A000               1660     LDY #0
59D4                    1680 DRAW05
59D4 A205               1700     LDX #5       ; #bytes/row
59D6                    1720 DRAW10
59D6 B1DE               1740     LDA (R+10),Y ; get current background
59D8 91E0               1760     STA (R+12),Y ; save
59DA 31E2               1780     AND (R+14),Y ; create "hole" with mask
59DC 11E4               1800     ORA (R+16),Y ; "OR" new data
59DE 91DE               1820     STA (R+10),Y ; write to screen
59E0 C8                 1840     INY
59E1 CA                 1860     DEX          ; dec count of bytes in row
59E2 D0F2               1880     BNE DRAW10
                        1900 ;
59E4 C6E7               1920     DEC R+19     ; dec count of rows to draw
59E6 F022               1940     BEQ DRAW20   ; branch when done
                        1960 ;
59E8 C6EA               1980     DEC R+22     ; time to wrap?
59EA D010               2000     BNE DRAW15   ; no - branch
59EC A5DE               2020     LDA R+10     ; else reset ptr
59EE 38                 2040     SEC          ; to buff beginning
59EF E95D               2060     SBC # <A112*40U-35
59F1 85DE               2080     STA R+10
59F3 A5DF               2100     LDA R+11
59F5 E911               2120     SBC # >A112*40U-35
59F7 85DF               2140     STA R+11
```

```
59F5 E911       2120     SBC # >A112*40U-35
59F7 85DF       2140     STA R+11
```



```
59F9 4CD459     2160     JMP DRAW05
59FC            2180 DRAW15
59FC A5DE       2200     LDA R+10
59FE 18         2220     CLC
59FF 6923       2240     ADC #35   ; inc ptr by 40 bytes/line - Yreg inc.
5A01 85DE       2260     STA R+10
5A03 90CF       2280     BCC DRAW05
5A05 E6DF       2300     INC R+11
5A07 4CD459     2320     JMP DRAW05
5A0A            2340 DRAW20
5A0A 18         2360     CLC       ; flag successful
5A0B            2380 DRAW30
5A0B 68         2400     PLA
5A0C AA         2420     TAX
5A0D 60         2440     RTS
                2460 ;
                2480 ;
                2500 ; ERASE ROUTINE - ERASE ENTRY IN
                2520 ; OBJTBL INDEXED BY Xreg
                2540 ;
5A0E            2560 ERASE
5A0E 8A         2562     TXA
5A0F 48         2564     PHA
5A10 BD7763     2580     LDA OVRLAY,X
5A13 C9FF       2600     CMP #$FF
5A15 F00B       2620     BEQ ERAS02
5A17 8A         2640     TXA
5A18 48         2660     PHA
5A19 BD7763     2680     LDA OVRLAY,X
5A1C AA         2700     TAX
5A1D 200E5A     2720     JSR ERASE
5A20 68         2740     PLA
5A21 AA         2760     TAX
5A22            2780 ERAS02
5A22 A900       2840     LDA #0    ; flag erase setup
5A24 20695A     2860     JSR DESETUP ; setup for erase
5A27 B03D       2880     BCS ERAS30  ; exit if not erasable
                2900 ;
5A29            2920 ERAS03
                2940 ;;  LDA YPOS,X  ; sync to screen
                2960 ;;  CLC
                2980 ;;  ADC #NSYNC2+17
                3000 ;;  SEC
                3020 ;;  SBC VCOUNT
                3040 ;;  BCC ERAS06
                3060 ;;  CMP #NSYNC
                3080 ;;  BCC ERAS03
5A29            3100 ERAS06
5A29 AD0BD4     3120     LDA VCOUNT
5A2C C96E       3140     CMP #110
5A2E D0F9       3160     BNE ERAS03
                3180 ;
                3200 ;;  LDA #$0F    ;; **DEBUG
                3220 ;;  STA $D01A   ;;
                3240 ;
5A30            3260 ERAS04
5A30 A000       3280     LDY #0
5A32 A910       3300     LDA #16    ; # rows to draw
```

Moore Business Forms, Inc. 19

```
5A30 A000       3280      LDY #0
5A32 A910       3300      LDA #16     ; # rows to draw
```



```
5A34 85E7       3320      STA R+19
5A36            3340 ERAS05
5A36 A205       3360      LDX #5
5A38            3380 ERAS10
5A38 B1E0       3400      LDA (R+12),Y ; get saved background
5A3A 91DE       3420      STA (R+10),Y ; restore to screen
5A3C C8         3440      INY
5A3D CA         3460      DEX
5A3E D0F8       3480      BNE ERAS10
                3500 ;
5A40 C6E7       3520      DEC R+19
5A42 F022       3540      BEQ ERAS30
                3560 ;
5A44 C6EA       3580      DEC R+22    ; time to wrap?
5A46 D010       3600      BNE ERAS15
5A48 A5DE       3620      LDA R+10
5A4A 38         3640      SEC
5A4B E95D       3660      SBC # <A112*40-35
5A4D 85DE       3680      STA R+10
5A4F A5DF       3700      LDA R+11
5A51 E911       3720      SBC # >A112*40-35
5A53 85DF       3740      STA R+11
5A55 4C365A     3760      JMP ERAS05
5A58            3780 ERAS15
5A58 A5DE       3800      LDA R+10
5A5A 18         3820      CLC
5A5B 6923       3840      ADC #35
5A5D 85DE       3860      STA R+10
5A5F 90D5       3880      BCC ERAS05
5A61 E6DF       3900      INC R+11
5A63 4C365A     3920      JMP ERAS05
5A66            3940 ERAS30
5A66 68         3960      PLA
5A67 AA         3980      TAX
                4000 ;;   LDA #$C6    ;; **DEBUG
                4020 ;;   STA $D01A   ;;
5A68 60         4040      RTS
                4060 ;
                4080 ;
                4100 ; DRAW/ERASE SETUP
                4120 ; SET REGISTERS WITH INFORMATION
                4140 ; NEEDED TO DO DRAW OR ERASE
                4160 ; IF Accum. = 0, ERASE IS SETUP
                4180 ; IF Accum.<> 0, DRAW IS SETUP
                4200 ;
                4220 ; R+10,11      PTR TO SCREEN
                4240 ; R+12,13      PTR TO BACKSAVE AREA
                4260 ; R+14,15      PTR TO MASK
                4280 ; R+16,17      PTR TO DATA
                4300 ; R+18         INIT. SCREEN ROW
                4320 ; R+20,21      USED
                4340 ; R+22         WRAPAROUND COUNT -
                4360 ;              # lines from end of
                4380 ;              buffer current ptr is
                4400 ; USES R+23
                4420 ;
5A69            4440 DESETUP
5A69 48         4460      PHA       ; save flag for draw or erase setup
```

LDA #0
STA DRAWN,X

Moore Business Forms, Inc. 1412RH

```
5A69 48          4440 DESETUP
                 4460      PHA          ; save flag for draw or erase setup
```



```
                 4480 ;
                 4500 ; Displayable?
                 4520 ;
5A6A BD2F63      4540      LDA YPOS,X   ; Y range is 0-95
5A6D C960        4560      CMP #96
5A6F 9003        4580      BCC DESE10
5A71 68          4600      PLA          ; remove flag from stack
5A72 38          4620      SEC          ; flag not successful
5A73 60          4640      RTS
                 4660 ;
                 4680 ; Set wraparound counter
                 4700 ;
5A74             4720 DESE10
5A74 18          4740      CLC
5A75 6D0840      4760      ADC DSCROL   ; Y is relative to top of screen
5A78 A8          4780      TAY
5A79 85EA        4800      STA R+22
5A7B A970        4820      LDA #112
5A7D 38          4840      SEC
5A7E E5EA        4860      SBC R+22
5A80 B002        4880      BCS DESE15
5A82 6970        4900      ADC #112
5A84             4920 DESE15
5A84 85EA        4940      STA R+22
                 4960 ;
                 4980 ; Set initial screen pointer
                 5000 ;
5A86 B93C40      5020      LDA DSPTBL,Y
5A89 85DE        5040      STA R+10
5A8B B91B41      5060      LDA DSPTBH,Y
5A8E 85DF        5080      STA R+11
5A90 BD2663      5100      LDA XPOS,X   ; get X
5A93 4A          5120      LSR A        ; convert to byte#
5A94 4A          5140      LSR A
5A95 18          5160      CLC
5A96 65DE        5180      ADC R+10     ; add to pointer
5A98 85DE        5200      STA R+10
5A9A 9002        5220      BCC DESE20
5A9C E6DF        5240      INC R+11
                 5260 ;
                 5280 ; Init backsave ptr
                 5300 ;
5A9E             5320 DESE20
5A9E 8A          5340      TXA          ; OBJTBL index determines backsave area
5A9F A8          5360      TAY
5AA0 B97F6B      5380      LDA BSTBLL,Y ; address low
5AA3 85E0        5400      STA R+12
5AA5 B9876B      5420      LDA BSTBLH,Y ;    " high
5AA8 85E1        5440      STA R+13
5AAA 68          5460      PLA          ; check draw or erase setup
5AAB F03F        5480      BEQ DESE30   ; if erase then done
                 5500 ;
                 5520 ; Init ptr to mask
                 5540 ;
5AAD BD2663      5560      LDA XPOS,X   ; form index to data/mask tables
5AB0 2903        5580      AND #$03     ; XPOS determines shift
5AB2 85EB        5600      STA R+23
5AB4 BD3863      5620      LDA DIR,X    ; dir determines image
```

Moore Business Forms, Inc. sv 14128H

```
5AB2 85EB       5600      STA R+23
5AB4 BD3863     5620      LDA DIR,X    ; dir determines image
```



```
5AB7 C902       5640      CMP #2       ; check for truck undef dir.
5AB9 9010       5660      BCC DESE57
5ABB F00C       5680      BEQ DESE56
5ABD 38         5700      SEC
5ABE E901       5720      SBC #1
5AC0 C904       5740      CMP #4
5AC2 9007       5760      BCC DESE57
5AC4 A902       5780      LDA #2
5AC6 4CCB5A     5800      JMP DESE57
5AC9            5820 DESE56
5AC9 A900       5840      LDA #0
5ACB            5860 DESE57
5ACB 0A         5880      ASL A
5ACC 0A         5900      ASL A
5ACD 18         5920      CLC
5ACE 65EB       5940      ADC R+23
5AD0 0A         5960      ASL A
5AD1 BC1D63     5980      LDY TYPE,X
5AD4 79C769     6000      ADC OBJREF,Y ; carry still clear
5AD7 A8         6020      TAY
5AD8 B9316A     6040      LDA OBJMSK,Y
5ADB 85E2       6060      STA R+14
5ADD B9326A     6080      LDA OBJMSK+1,Y
5AE0 85E3       6100      STA R+15
                6120 ;
                6140 ; Init ptr to data
                6160 ;
5AE2 B9F169     6180      LDA OBJDAT,Y ; data addr. high
5AE5 85E4       6200      STA R+16
5AE7 B9F269     6220      LDA OBJDAT+1,Y ; "  "  low
5AEA 85E5       6240      STA R+17
                6260 ;
5AEC            6280 DESE30
5AEC 18         6300      CLC          ; flag successful
5AED 60         6320      RTS
                6340 ;
                6360 ; PAUSE WHEN SPACE BAR PRESSED.
                6380 ; WAIT UNTIL ANY KEYPRESS TO
                6400 ; CONTINUE
                6420 ;
5AEE            6440 PAUSE
5AEE ADFC02     6460      LDA CH       ; check keyboard
5AF1 C921       6480      CMP #$21     ; space bar
5AF3 D019       6500      BNE PAUS50   ; exit if not pressed
5AF5 A200       6520      LDX #0       ; delay
5AF7 A000       6540      LDY #0
5AF9            6560 PAUS10
5AF9 CA         6580      DEX
5AFA D0FD       6600      BNE PAUS10
5AFC 88         6620      DEY
5AFD D0FA       6640      BNE PAUS10
                6660 ;
5AFF A9FF       6680      LDA #$FF
5B01 8DFC02     6700      STA CH
5B04 8DF202     6720      STA CH1
5B07            6740 PAUS20
5B07 AD01D2     6760      LDA $D201
5B0A C921       6780      CMP #$21     ; wait for release
```

Moore Business Forms, Inc sv 14128M

```
                                    ; wait for release
```



```
5B0C F0F9       6800      BEQ PAUS20
5B0E            6820 PAUS50
5B0E 60         6840      RTS
                014460    .INCLUDE #D2:TRACK3.M65
                0900      .PAGE "TRACK"
```

Moore Business Forms, Inc. sv 14128H

```
TRACK                                   D2:TRACK3.M65

                    1000 ; TRACK3.M65
                    1020 ;
      =00QB         1040 YTRKPT = 11    ; Y track point relative to top of object
5B0F                1060 BLSTINC *= *+1
5B10                1080 BLSTNZR *= *+1
5B11                1100 BUGDIR *= *+1  ; player's direction
5B12                1120 BPTHCT *= *+1  ; index to path player is following
5B13                1140 SCRFLG *= *+1  ; flag set when scrolling occurs
      =0006         1160 JMPDLY = 6     ; delay during jump sequencing
5B14                1180 JMPCTR *= *+1  ; count for delay between jump moves
5B15                1200 OLDBUGY *= *+1 ; player's last Y position
5B16                1220 OLDBDIR *= *+1 ; player's last direction
      =D300         1240 PORTA = $D300  ; joystick port
      =D302         1260 PACTL = $D302  ; PORTA control
      =0010         1280 FWDSPD = 16    ; max fwd. speed (must be positive)
      =FFF0         1300 BWDSPD = -16   ; max bwd. speed (must be negative)
      =0028         1320 ACCDLY = 40    ; delay betw. accel. incs.
5B17                1340 ACCCNT *= *+1  ; counter for accel delay
5B18                1360 JMPNDX *= *+1  ; index to jump path
      =D010         1380 TRIGO = $D010  ; joystick trigger
      =0008         1400 NPATH = 8      ; # path iterations to get across
                                        intersection
5B19                1420 WHLSEQ *= *+1  ; sequence count for wheel spin
                    1440 ;
                    1460 ;
                    1480 ; ROUTINE RETURNS X FOR A VEHICLE
                    1500 ; SUCH THAT VEHICLE TRACKS THE ROAD
                    1520 ;
                    1540 ; ENTER WITH:
                    1560 ;     R+0     present X (0-159)
                    1580 ;     R+1     new Y     (0-95)
                    1600 ;     R+2     present direction
                    1620 ;
                    1640 ;               left        right
                    1660 ;                0    level   3
                    1680 ;                1    up      4
                    1700 ;                2    down    5
                    1720 ;     R+10    last increment
                    1740 ;     R+11    last non-zero incr.
                    1760 ;     R+12    speed (negative if in reverse)
                    1780 ;     R+8     forced path counter
                    1800 ; RETURNS:
                    1820 ;     R+0     new X
                    1840 ;     R+3     road status byte
                    1860 ;             0-5   direction
                    1880 ;             128 = hit ground
                    1900 ;             129 = hit water
                    1920 ;             bit6=1, turning point
                    1940 ; USES:
                    1960 ;     R+4,5,6,7,9
5B1A                1980 TRACK
5B1A 98             2000     TYA
5B1B 48             2020     PHA
5B1C 8A             2040     TXA
5B1D 48             2060     PHA
5B1E                2080 TRACO2
```

Moore Business Forms, Inc. av 1412RH

```
                    2140 ;
                    2160 ; On path - get next byte and add
                    2180 ; as Xpos. increment
                    2200 ;
5B22 2903           2220     AND #3         ; calculate path index
5B24 85DD           2240     STA R+9
5B26 A5D6           2260     LDA R+2
5B28 0A             2280     ASL A
5B29 0A             2300     ASL A
5B2A 05DD           2320     ORA R+9
5B2C AA             2340     TAX            ; path index in Xreg
5B2D BD5C5C         2360     LDA PATHD,X    ; get inc.
5B30 A6E0           2400     LDX R+12       ; forward or backward?
5B32 1005           2420     BPL TRAC05
5B34 49FF           2440     EOR #$FF       ; invert addend
5B36 18             2460     CLC
5B37 6901           2480     ADC #1
5B39                2500 TRAC05
5B39 85DE           2502     STA R+10       ; save as last inc.
5B3B 0900           2503     ORA #0         ; set zero flag
5B3D F002           2504     BEQ TRAC06
5B3F 85DF           2506     STA R+11       ; save as last non-zero
5B41                2508 TRAC06
5B41 18             2520     CLC            ; add to Xpos.
5B42 65D4           2540     ADC R+0
5B44 85D4           2560     STA R+0
                    2580 ;
5B46 C6DC           2600     DEC R+8        ; update index
5B48 4CE15B         2620     JMP TRAC70
                    2640 ;
                    2660 ; Track the left edge of the road
                    2680 ;
5B4B                2700 TRAC10
5B4B A5D5           2720     LDA R+1        compute ptr to screen from X,Y
5B4D 18             2740     CLC
5B4E 6D0840         2760     ADC DSCROL
5B51 690B           2780     ADC #YTRKPT
5B53 A8             2800     TAY
5B54 A5D4           2820     LDA R+0        ; get x
5B56 4A             2840     LSR A          ; find horizontal byte offset
5B57 4A             2860     LSR A
5B58 18             2880     CLC
5B59 793C40         2900     ADC DSPTBL,Y ; low byte
                    2920 ;
5B5C 85DA           2940     STA R+6
5B5E B91B41         2960     LDA DSPTBH,Y ; high byte
5B61 6900           2980     ADC #0
5B63 85DB           3000     STA R+7
                    3020 ;
5B65 A5DA           3040     LDA R+6        ; -5 to allow search of left byte
5B67 38             3060     SEC
5B68 E905           3080     SBC #5
5B6A 85DA           3100     STA R+6
5B6C B002           3120     BCS TRAC18
5B6E C6DB           3140     DEC R+7
                    3160 ;
5B70                3180 TRAC18
5B70 A202           3200     LDX #2         ; # bytes to search-1
```

Moore Business Forms, Inc. 14128M

```
5B70 A202       3200     LDX #2       ; # bytes to search-1


PAGE 30   UPN DOWN - for Atari systems
TRACK                                                    D2:TRACK3.M65

5B72 A006       3220     LDY #6       ; rightmost byte to search (middle is #5)
5B74            3240 TRAC20
5B74 B1DA       3260     LDA (R+6),Y ; find a non-road byte
5B76 C9FF       3280     CMP #$FF
5B78 D00B       3300     BNE TRAC30
                3320 ;   LDA #$1B
                3340 ;   STA (R+6),Y
5B7A 88         3360     DEY
5B7B CA         3380     DEX
5B7C 10F6       3400     BPL TRAC20
                3420 ;
                3440 ; search failed
                3460 ;
5B7E A908       3580     LDA #NPATH  ; set on path according to current
                                     direction
5B80 85DC       3600     STA R+8
5B82 4C1E5B     3620     JMP TRAG02
                3640 ;
5B85            3660 TRAC30
5B85 48         3680     PHA         ; save left road byte
5B86 A5D4       3700     LDA R+0
5B88 29FC       3720     AND #$FC
5B8A 18         3740     CLC
5B8B 7D595C     3760     ADC BYTADD,X ; add byte offset to new X
5B8E 85DD       3780     STA R+9
                3800 ;
5B90 68         3820     PLA
5B91 A203       3840     LDX #3
5B93            3860 TRAC40
5B93 6A         3880     ROR A       ; find pixel offset of L side of road
5B94 9006       3900     BCC TRAC50
5B96 6A         3920     ROR A
5B97 9003       3940     BCC TRAC50
5B99 CA         3960     DEX
5B9A D0F7       3980     BNE TRAC40
                4000 ;
5B9C            4020 TRAC50
                4060 ;   ORA BB,X
5B9C 8A         4100     TXA
5B9D 18         4120     CLC
5B9E 65DD       4140     ADC R+9
5BA0 85DD       4160     STA R+9
5BA2 38         4180     SEC
5BA3 E5D4       4200     SBC R+0
5BA5 85D9       4220     STA R+5     ; save increment
5BA7 A5DD       4240     LDA R+9     ; set new X
5BA9 85D4       4260     STA R+0
5BAB A6D4       4280     LDX R+0
5BAD A4D5       4300     LDY R+1
                4320 ;
                4340 ; set new direction
                4360 ;
5BAF A5D9       4380     LDA R+5     ; get inc
5BB1 A6E0       4400     LDX R+12    ; fwd or bwd?
5BB3 1007       4420     BPL TRAC48
5BB5 49FF       4440     EOR #$FF
5BB7 18         4460     CLC
```

Moore Business Forms, Inc. sv 1412RH

```
5BB8 6901       4480      ADC #1

PAGE 31   UPN DOWN - for Atari systems
TRACK                                                    D2:TRACK3.M65

5BBA 85D9       4500      STA R+5
5BBC            4520 TRAC48
5BBC AA         4540      TAX
5BBD E4DE       4580      CPX R+10    ; cmp to last inc
5BBF D00B       4600      BNE TRAC56
5BC1 A5DE       4602      LDA R+10
5BC3 F012       4604      BEQ TRAC60
5BC5 E8         4620      INX
5BC6 E8         4640      INX
5BC7 BD535C     4660      LDA DIRINC,X
5BCA 85D6       4680      STA R+2
5BCC            4700 TRAC56
5BCC A5D9       4720      LDA R+5
5BCE 85DE       4740      STA R+10    ; save as last inc
5BD0 F00F       4742      BEQ TRAC65
5BD2 85DF       4760      STA R+11    ; save as last non-zero
5BD4 4CE15B     4780      JMP TRAC65
5BD7            4800 TRAC60
5BD7 A204       4802      LDX #4
5BD9 A5DF       4860      LDA R+11
5BDB 1002       4880      BPL TRAC64
5BDD A201       4900      LDX #1
5BDF            4980 TRAC64
5BDF 86D6       5000      STX R+2
5BE1            5020 TRAC65
                5160 ;
                5180 ; Calculate pointer to status byte
                5200 ;
5BE1            5220 TRAC70
5BE1 A5D4       5240      LDA R+0     ; ptr=SCROLC+A(Y-32)/320*10+(X+4)/16
5BE3 18         5260      CLC         ; offset X
5BE4 6908       5280      ADC #8
5BE6 4A         5300      LSR A
5BE7 4A         5320      LSR A
5BE8 4A         5340      LSR A
5BE9 4A         5360      LSR A
5BEA 85D8       5380      STA R+4
5BEC A5D5       5420      LDA R+1     ; get Y
5BEE 18         5440      CLC
5BEF 690B       5460      ADC #YTRKPT
5BF1 18         5480      CLC
5BF2 6D0B40     5500      ADC SCROLF
5BF5 4A         5520      LSR A
5BF6 4A         5540      LSR A
5BF7 4A         5560      LSR A
5BF8 4A         5580      LSR A
5BF9 A8         5600      TAY
5BFA B9244B     5620      LDA M10TBL,Y ; *10
5BFD 18         5640      CLC
5BFE 65D8       5660      ADC R+4     ; won't ovfl
5C00 65CA       5680      ADC SCROLC
5C02 85D8       5700      STA R+4
5C04 A900       5720      LDA #0
5C06 65CB       5740      ADC SCROLC+1
5C08 85D9       5760      STA R+5
                5780 ;
                5800 ; Correct for overflow
                5820 ;
```

Moore Business Forms, Inc. sv 1412RH

```
5820 ;


PAGE 32   UPN DOWN - for Atari systems
TRACK                                                    D2:TRACK3.M65


5C0A A5D9       5840     LDA R+5
5C0C C946       5860     CMP # >APFDATA+PFLENU
5C0E 9015       5880     BCC TRAC90
5C10 D006       5900     BNE TRAC80
5C12 A5D8       5920     LDA R+4
5C14 C920       5940     CMP # <APFDATA+PFLENU
5C16 900D       5960     BCC TRAC90
5C18            5980 TRAC80
5C18 A5D8       6000     LDA R+4
5C1A 38         6020     SEC
5C1B E96C       6040     SBC # <PFLEN
5C1D 85D8       6060     STA R+4
5C1F A5D9       6080     LDA R+5
5C21 E902       6100     SBC # >PFLEN
5C23 85D9       6120     STA R+5
                6140 ;
                6160 ; Get road status byte
                6180 ;
5C25            6200 TRAC90
5C25 A000       6220     LDY #0
5C27 B1D8       6240     LDA (R+4),Y ; get char #
5C29 AA         6280     TAX
5C2A BD915E     6300     LDA RDSTAT1,X ; get byte from table
5C2D 85D7       6320     STA R+3
                6340 ;
                6360 ; Turn point?
                6380 ;
5C2F 2940       6400     AND #$40    ; check for intrsctn.
5C31 F01B       6420     BEQ TRA100
5C33 A5D5       6440     LDA R+1
5C35 18         6460     CLC
5C36 6D0B40     6480     ADC SCROLF
5C39 690B       6500     ADC #YTRKPT ; won't ovfl
5C3B 290F       6520     AND #$0F
5C3D 85D9       6540     STA R+5
5C3F BDBF5E     6560     LDA RDSTAT2,X ;check which line turning pt. strobes
5C42 290F       6580     AND #$0F
5C44 C5D9       6600     CMP R+5
                6620 ;
5C46 F006       6640     BEQ TRA100  ; yes - branch
5C48 A5D7       6660     LDA R+3     ; else no turns
5C4A 29BF       6680     AND #$BF
5C4C 85D7       6700     STA R+3
5C4E            6720 TRA100
5C4E 68         6740     PLA
5C4F AA         6760     TAX
5C50 68         6780     PLA
5C51 A8         6800     TAY
5C52 60         6820     RTS
5C53 02000103   6860 DIRINC .BYTE 2,0,1,3,5,0
5C57 0500
                6880 ;
                6900 ; Values to add to Xpos. to
                6920 ; move 1 byte  left, nowhere, right
                6940 ;
5C59 FC0004     6960 BYTADD .BYTE -4,0,4
                6980 ;
                7000 ; Paths for "lost" vehicles are
```

```
                        6980 ;
                        7000 ; Paths for "lost" vehicles are
```



```
                        7020 ; 4 bytes long and wrap around
                        7040 ;
5C5C                    7060 PATHD
5C5C FFFFFFFF           7080      .BYTE -1,-1,-1,-1 ; dir. =0
5C60 FF000000           7100      .BYTE -1,0,0,0 ;     1
5C64 FEFEFEFE           7120      .BYTE -2,-2,-2,-2 ;     2
5C68 01010101           7140      .BYTE 1,1,1,1 ;     3
5C6C 01000000           7160      .BYTE 1,0,0,0 ;     4
5C70 02020202           7180      .BYTE 2,2,2,2 ;     5 .
                        7200 ;
                        7220 ; Jump path
                        7240 ;
5C74                    7260 JMPADD ;       used to calc. Y during jump
5C74 FEFCFAF8           7280      .BYTE -2,-4,-6,-8,-9,-11,-12,-14,-16,-17
5C78 F7F5F4F2
5C7C F0EF
5C7E EEEDECEB           7300      .BYTE -18,-19,-20,-21,-21,-22,-23,-23,-24,-24
5C82 EBEAE9E9
5C86 E8E8
5C88 E8E8E8E8           7320      .BYTE -24,-24,-24,-24,-24,-24,-24,-23,-23,-22,-21,-21
5C8C E8E8E8E9
5C90 E9EAEBEB
5C94 ECEDEEEF           7340      .BYTE -20,-19,-18,-17,-16,-14,-12,-11,-9,-8,-6,-4,-2
5C98 F0F2F4F5
5C9C F7F8FAFC
5CA0 FE
5CA1                    7360 JMPPIC ;       used to calc pic during jump
5CA1 01010101           7380      .BYTE 1,1,1,1,1,1,1,1,1,1
5CA5 01010101
5CA9 0101
5CAB 01010101           7400      .BYTE 1,1,1,1,1,1,1,1,0,0,0
5CAF 01010100
5CB3 0000
5CB5 00000000           7420      .BYTE 0,0,0,0,0,0,0,0,2,2,2,2
5CB9 00000000
5CBD 02020202
5CC1 02020202           7440      .BYTE 2,2,2,2,2,2,2,2,2,2,2,0,0,0
5CC5 02020202
5CC9 02020200
5CCD 0000
     =002D              7460 ENDJMP = JMPPIC-JMPADD ; # bytes in each of the above
                                    lists
                        7480 ;
                        7500 ;
                        7520 ; MOVE THE PLAYERS 'BUG'
                        7540 ;
5CCF                    7560 PMOVE
5CCF ADA642             7580      LDA BUGX     ; player's X
5CD2 85D4               7600      STA R+0
5CD4 ADA742             7620      LDA BUGY     ;      Y
5CD7 85D5               7640      STA R+1
5CD9 AD0F5B             7660      LDA BLSTINC
5CDC 85DE               7680      STA R+10
5CDE AD105B             7700      LDA BLSTNZR
5CE1 85DF               7720      STA R+11
5CE3 AD115B             7740      LDA BUGDIR  ;     direction
5CE6 85D6               7760      STA R+2
5CE8 AD0A40             7780      LDA SCRSPD
```

Moore Business Forms, Inc. 1412RH

```
5CE8 85D8       7760      STA R+2
5CE8 AD0A40     7780      LDA SCRSPD
```



```
5CEB 85E0       7800      STA R+12
                7820 ;
5CED AD125B     7840      LDA BPTHCT  ;     path count
5CF0 85DC       7860      STA R+8
                7880 ;
                7900 ; Acceleration/deceleration
                7920 ;
5CF2            7940 PMOV10
5CF2 CE175B     7960      DEC ACCCNT
5CF5 D028       7980      BNE PMOV30
5CF7 A928       8000      LDA #ACCDLY
5CF9 8D175B     8020      STA ACCCNT
5CFC AD00D3     8040      LDA PORTA   ; check joystick
5CFF 2903       8060      AND #3      ; ... for forward/backward pos.
5D01 C903       8080      CMP #3
5D03 F01A       8100      BEQ PMOV30  ; neither selected
5D05 2901       8120      AND #1      ; stick forward?
5D07 D00B       8140      BNE PMOV20  ; no - branch
5D09 AE0A40     8160      LDX SCRSPD  ; accelerate
5D0C E8         8180      INX
                8200 ;    CPX #128    ; can't go negative
5D0D E010       8220      CPX #FWDSPD
5D0F F00E       8240      BEQ PMOV30
5D11 4C1C5D     8260      JMP PMOV25
5D14            8280 PMOV20
5D14 AE0A40     8300      LDX SCRSPD  ;decelerate
5D17 CA         8320      DEX
                8340 ;    CPX #127    ; can't go positive
5D18 E0F0       8360      CPX #BWDSPD
5D1A F003       8380      BEQ PMOV30
5D1C            8400 PMOV25
5D1C 8E0A40     8420      STX SCRSPD
                8440 ;
                8460 ; JUMPS
                8480 ;
5D1F            8500 PMOV30
5D1F AE185B     8520      LDX JMPNDX  ; jump in progress?
5D22 101B       8540      BPL PMOV33  ; yes - branch
                8560 ;
5D24 AD10D0     8580      LDA TRIG0   ; check trigger button
5D27 F003       8600      BEQ PMOV32  ; branch if pressed
5D29 4CDC5D     8620      JMP PMOV60
5D2C            8640 PMOV32
5D2C AD0A40     8660      LDA SCRSPD  ; can only jump if not moving backwards
5D2F 1003       8680      BPL PMOV31
5D31 4CDC5D     8700      JMP PMOV60
5D34            8720 PMOV31
5D34 A200       8740      LDX #0      ; start the jump
5D36 8E185B     8760      STX JMPNDX
5D39 20CC65     8780      JSR VWJUMP  ; do jump sound
5D3C 4CB95D     8800      JMP PMOV40
                8820 ;
                8840 ; Update X during jump
                8860 ;
5D3F            8880 PMOV33
5D3F AD135B     8900      LDA SCRFLG
5D42 F035       8920      BEQ PMOV34  ; no X change
5D44            8940 PMO330
```

```
5D44                8940 PM0330
```



```
5D44 201A5B     8960     JSR TRACK
5D47 A5D7       8980     LDA R+3     ; check for jumping over intrsctn
5D49 2940       9000     AND #$40
5D4B F007       9020     BEQ PM0332
5D4D A908       9040     LDA #NPATH  ; force path if intrsctn
5D4F 85DC       9060     STA R+8
5D51 4C795D     9080     JMP PMOV34
5D54            9100 PM0332
5D54 A5DF       9120     LDA R+11    ; check for change of dir. while jumping
5D56 C904       9160     CMP #4
5D58 9004       9180     BCC PM0333  ; check for illegal track
5D5A C9FD       9200     CMP #-3
5D5C 9007       9220     BCC PM0334
5D5E            9240 PM0333
5D5E AD105B     9260     LDA BLSTNZR
5D61 45DF       9300     EOR R+11    ; sign change?
5D63 1014       9320     BPL PMOV34  ; no, ok to track - branch
5D65            9340 PM0334
5D65 AD115B     9360     LDA BUGDIR  ; else keep same dir.
5D68 85D6       9380     STA R+2
5D6A ADA642     9400     LDA BUGX
5D6D 85D4       9420     STA R+0
5D6F A901       9440     LDA #1
5D71 85DC       9460     STA R+8
5D73 201A5B     9480     JSR TRACK
5D76 4C7E5D     9500     JMP PM0342
5D79            9520 PMOV34
5D79 A5D6       9560     LDA R+2
5D7B 8D115B     9600     STA BUGDIR
                9620 ;
                9640 ; Sequence jump pictures
                9660 ;
5D7E            9680 PM0342
5D7E CE145B     9700     DEC JMPCTR  ; delay between jump moves
5D81 F003       9720     BEQ PMOV35
5D83 4CB95D     9740     JMP PMOV40
5D86            9760 PMOV35
5D86 A906       9780     LDA #JMPDLY
5D88 8D145B     9800     STA JMPCTR
5D8B E8         9820     INX         ; inc for"next jump step
5D8C E02D       9840     CPX #ENDJMP ; end of sequence?
5D8E D026       9860     BNE PMOV45  ; no - branch
5D90 A280       9880     LDX #128    ; else cancel jumr"index
5D92 8E185B     9900     STX JMPNDX
                9920 ;
                9940 ; Check for killing a truck
                9960 ;
5D95 208965     9980     JSR COLISN
5D98 B003       010000   BCS PMOV42
5D9A 4C905E     010020   JMP PMOV90  ; jump if no hit
5D9D            010040 PMOV42
5D9D 20DE5A     010060   JSR ERASE   ; erase the truck
5DA0 A9FF       010080   LDA #-1     ; and cancel it
5DA2 9D1D63     010100   STA TYPE,X
5DA5 20DC65     010120   JSR TRKDIE  ; do truck die sound
                010140 ;
                010160 ; bonus for killing truck
                010180 ;
```

Moore Business Forms, Inc. sv 1412RH

```
                010160 ; bonus for killing truck
                010180 ;


PAGE 36   UPN DOWN - for Atari systems
TRACK                                                    D2:TRACK3.M65

5DA8 A900       010200   LDA #0       ; add 400
5DAA 85D9       010220   STA R+5
5DAC A904       010240   LDA #$04
5DAE 85DA       010260   STA R+6
5DB0 20EE5F     010280   JSR ADDP1    ; add to score
5DB3 4C905E     010300   JMP PMOV90
                010320 ;
                010340 ; Update jump picture/position
                010360 ;
5DB6            010380 PMOV45
5DB6 8E185B     010400   STX JMPNDX
5DB9            010402 PMOV40
5DB9 A5D5       010440   LDA R+1      ; modify player's Y
5DBB 18         010460   CLC
5DBC 7D745C     010480   ADC JMPADD,X
5DBF A8         010500   TAY          ; save
5DC0 BDA15C     010520   LDA JMPPIC,X ; get picture
5DC3 AE115B     010540   LDX BUGDIR   ; modify for left/right
5DC6 E003       010560   CPX #3
5DC8 9003       010580   BCC PMOV46
5DCA 18         010600   CLC
5DCB 6903       010620   ADC #3
5DCD            010640 PMOV46
5DCD 85DD       010660   STA R+9      ; set dir to display
5DCF A6D4       010680   LDX R+0
5DD1 8EA642     010700   STX BUGX     ; update player's X
5DD4 A5DC       010720   LDA R+8
5DD6 8D125B     010740   STA BPTHCT
5DD9 4C835E     010772   JMP PMOV80
                010780 ;
                010800 ; get next X position
                010820 ;
5DDC            010840 PMOV60
5DDC AD135B     010860   LDA SCRFLG  ; X only changes after scroll
5DDF D003       010880   BNE PMOV611
5DE1 4C6B5E     010882   JMP PMOV70  ; no turns if no scroll
5DE4            010884 PMOV611
5DE4 201A5B     010900   JSR TRACK
                010920 ;
                010940 ; TURNS
                010960 ;
5DE7            010980 PMOV61
5DE7 A5D7       011000   LDA R+3      ; check road status (returned from TRACK
                                      call)
5DE9 2940       011020   AND #$40     ; intersection?
5DEB F07E       011040   BEQ PMOV70   ; branch if not a turning point
5DED AD00D3     011060   LDA PORTA
5DF0 290C       011080   AND #$0C
5DF2 C90C       011100   CMP #$0C
5DF4 D02A       011120   BNE PM0628   ; branch if turning
                011140 ;
                011160 ; coast across intersection
                011180 ;
5DF6 AD0A40     011200   LDA SCRSPD   ; special case is dir 3, fwd ...
5DF9 300F       011220   BMI PM0625   ; check fwd
5DFB A5D7       011240   LDA R+3      ; check can turn
5DFD 2902       011260   AND #UR
5DFF F009       011280   BEQ PM0625
```

Moore Business Forms, Inc. s/y 14128H

```
5DFD 2902       011260   AND #UR
5DFF F009       011280   BEQ PM0625


PAGE 37   UPN DOWN - for Atari systems
TRACK                                                    D2:TRACK3.M65

5E01 A5D6       011300   LDA R+2     ; check dir
5E03 C903       011320   CMP #3
5E05 D003       011340   BNE PM0625
5E07 4C615E     011360   JMP PMOV68  ; force turn
5E0A            011380 PM0625
5E0A AD0A40     011400   LDA SCRSPD  ; ... or dir 0, bwd
5E0D 105C       011420   BPL PMOV70
5E0F A5D7       011440   LDA R+3
5E11 2908       011460   AND #LR
5E13 F056       011480   BEQ PMOV70
5E15 A5D6       011500   LDA R+2
5E17 D052       011520   BNE PMOV70
5E19 E6D4       011540   INC R+0     ; fudge to align X
5E1B E6D4       011560   INC R+0
5E1D 4C615E     011580   JMP PMOV68  ; force turn
                011600 ;
5E20            011620 PM0628
5E20 2908       011640   AND #8      ; turning right?
5E22 D025       011660   BNE PMOV64  ; no - branch
5E24 AD0A40     011680   LDA SCRSPD  ; going forward or backward?
5E27 300B       011700   BMI PMOV62
5E29 A5D7       011720   LDA R+3     ; can make this turn?
5E2B 2902       011740   AND #UR
5E2D F03C       011760   BEQ PMOV70  ; no - branch
5E2F A903       011780   LDA #3      ; forward, right
5E31 4C615E     011800   JMP PMOV68
5E34            011820 PMOV62
5E34 A5D7       011840   LDA R+3
5E36 2908       011860   AND #LR
5E38 F031       011880   BEQ PMOV70
5E3A A5D6       011900   LDA R+2     ; turning right, bwd needs fudge
                                     correction
5E3C C903       011920   CMP #3
5E3E D004       011940   BNE PMOV63
5E40 E6D4       011960   INC R+0
5E42 E6D4       011980   INC R+0
5E44            012000 PMOV63
5E44 A900       012020   LDA #0      ; backward, right
5E46 4C615E     012040   JMP PMOV68
5E49            012060 PMOV64
5E49 AD0A40     012080   LDA SCRSPD
5E4C 300B       012100   BMI PMOV66
5E4E A5D7       012120   LDA R+3
5E50 2901       012140   AND #UL
5E52 F017       012160   BEQ PMOV70
5E54 A900       012180   LDA #0      ; forward, left
5E56 4C615E     012200   JMP PMOV68
5E59            012220 PMOV66
5E59 A5D7       012240   LDA R+3
5E5B 2904       012260   AND #LL
5E5D F00C       012280   BEQ PMOV70
5E5F A903       012300   LDA #3      ; backward, left
5E61            012320 PMOV68
5E61 85D6       012340   STA R+2     ; change direction
5E63 A908       012360   LDA #NPATH  ; force path to get across intersection
5E65 8D125B     012380   STA BPTHCT
5E68 4C705E     012400   JMP PMOV75
5E6B            012420 PMOV70
```

Moore Business Forms, Inc. av 1412RH

```
5E68 4C705E     012400   JMP PMOV75
5E6B            012420 PMOV70
```



```
5E6B A5DC       012440   LDA R+8
5E6D 8D125B     012460   STA BPTHCT
5E70            012480 PMOV75
5E70 A5D4       012500   LDA R+0     ; update X
5E72 8DA642     012520   STA BUGX
5E75 AA         012540   TAX
5E76 A5D5       012560   LDA R+1
5E78 8DA742     012580   STA BUGY
5E7B A8         012600   TAY
5E7C A5D6       012620   LDA R+2
5E7E 8D115B     012640   STA BUGDIR
5E81 85DD       012650   STA R+9
5E83            012658 PMOV80
5E83 A5DE       012660   LDA R+10
5E85 8D0F5B     012680   STA BLSTINC
5E88 A5DF       012700   LDA R+11
5E8A 8D105B     012720   STA BLSTNZR
                012740   ;
5E8D 20ED5E     012780   JSR SHOCAR  ; display VW at new position
                012800 ;
5E90            012820 PMOV90
5E90 60         012900   RTS
                012920 ;
                012940 ; TABLE OF STATUS BYTES FOR ROAD
                012960 ; CHARACTERS
                012980 ; bits:
                013000 ;    Road choices
                013020 ; 0  Upper Left
                013040 ; 1  Upper Right
                013060 ; 2  Lower Left
                013080 ; 3  Lower Right
                013100 ;
                013120 ; 4,5  grade
                013140 ;        = 00 flat
                013160 ;        = 01 uphill
                013180 ;        = 10 downhill
                013200 ; 6    intersection
                013220 ;        = 0   no intersct.
                013240 ;        = 1   intersection
                013260 ; 7    flags a char. which
                013280 ;      vehicles cannot be on
                013300 ;      Then if bit 0 =0 , ground
                013320 ;                =1 , water
                013340 ;
5E91            013360 RDSTAT1
5E91 80000003   013380   .BYTE 128,0,0,3,3,$43,$4C,129
5E95 03434C81
5E99 81808080   013400   .BYTE 129,128,128,128,128,128,0,0
5E9D 80800000
5EA1 0303434C   013420   .BYTE 3,3,$43,$4C,16+1,16+1,16+1,16+1
5EA5 11111111
5EA9 80801111   013440   .BYTE 128,128,16+1,16+1,16+1,128,32+2,32+2
5EAD 11802222
5EB1 22252525   013460   .BYTE 32+2,32+5,32+5,32+5,128,128,128,128
5EB5 80808080
5EB9 80808080   013480   .BYTE 128,128,128,128,128,128
5EBD 8080
                013500 ; Second table indicates which line
```

Moore Business Forms, Inc. sv 1412RH

```
                        013500 ; Second table indicates which line
                        013520 ; of char. turning pt. occurs
5EBF                    013540 RDSTAT2
5EBF 00000000           013560    .BYTE 0,0,0,0,0,15,0,0
5EC3 000F0000
5EC7 00000000           013580    .BYTE 0,0,0,0,0,0,0,0
5ECB 00000000
5ECF 00000F00           013600    .BYTE 0,0,15,0,0,0,0,0
5ED3 00000000
5ED7 00000000           013620    .BYTE 0,0,0,0,0,0,0,0
5EDB 00000000
5EDF 00000000           013640    .BYTE 0,0,0,0,0,0,0,0
5EE3 00000000
5EE7 00000000           013660    .BYTE 0,0,0,0,0,0
5EEB 0000
                        013680 ;
      =0001             013700 UL =  $01
      =0002             013720 UR =  $02
      =0004             013740 LL =  $04
      =0008             013760 LR =  $08
                        013780 ;
                        013800 ;
                        013820 ;
                        013840 ;
                        013860 ; DRAW THE CAR AS A P/M GRAPHIC
                        013880 ; Xreg and Yreg hold car's X,Y
5EED                    013900 SHOCAR
                        013910 ; Direction to display is R+9
                        013912 ;
5EED 8A                 013920    TXA         ; convert to P/M  X coord.
5EEE 18                 013940    CLC
5EEF 6930               013970    ADC #48
5EF1 8D1860             013980    STA BUGH0   ; set X
5EF4 8D1960             014000    STA BUGH1
5EF7 6908               014020    ADC #8
5EF9 8D1A60             014040    STA BUGH2
5EFC 8D1B60             014060    STA BUGH3
                        014080 ;
                        014100 ; Update Y
                        014120 ;
5EFF 98                 014140    TYA         ; convert to P/M Y coord.
5F00 0A                 014160    ASL A
5F01 18                 014180    CLC
                        014420 ;
5F02 6920               014440    ADC #32
5F04 A8                 014460    TAY
5F05 CC155B             014480    CPY OLDBUGY ; change in Y?
5F08 D00A               014500    BNE SHOC03  ; yes - erase and redraw
5F0A A5DD               014520    LDA R+9     ; change in direction?
5F0C CD165B             014540    CMP OLDBDIR
5F0F D02D               014560    BNE SHOC07  ; yes - redraw but don't erase
5F11 4C725F             014580    JMP SHOC20  ; exit without rewriting P/M
                        014600 ;
                        014620 ; erase old BUG
                        014640 ;
5F14                    014660 SHOC03
5F14 98                 014680    TYA
5F15 AC155B             014700    LDY OLDBUGY
5F18 8D155B             014720    STA OLDBUGY ; save new Y pos
```

Moore Business Forms, Inc. sv 14128H

```
5F15 AC155B     014700   LDY OLDBUGY
5F18 8D155B     014720   STA OLDBUGY ; save new Y pos
```



```
5F1B A900       014740   LDA #0
5F1D A205       014760   LDX #5       ; erase 5 top andbottom byte of old image
5F1F            014780 SHOC05
5F1F 990074     014800   STA P0,Y
5F22 991B74     014810   STA P0+27,Y
5F25 990075     014820   STA P1,Y
5F28 991B75     014830   STA P1+27,Y
5F2B 990076     014840   STA P2,Y
5F2E 991B76     014850   STA P2+27,Y
5F31 990077     014860   STA P3,Y
5F34 991B77     014870   STA P3+27,Y
5F37 C8         014880   INY
5F38 CA         014900   DEX
5F39 D0E4       014920   BNE SHOC05
5F3B AC155B     014940   LDY OLDBUGY ; restore new Y
5F3E            014960 SHOC07
                014980 ;
5F3E A5D9       015000   LDA R+9      ; form index to VW P/M data
5F40 C906       015020   CMP #6       ; check dir.
5F42 9003       015039   BCC SHOC08
5F44 4C755F     015040   JMP SHOC30   ; don't draw invalid dir.
5F47            015100 SHOC08
5F47 8D165B     015120   STA OLDBDIR
5F4A AA         015140   TAX
5F4B BD765F     015160   LDA M32TBL,X ; *32
5F4E AA         015180   TAX
5F4F 18         015200   CLC
5F50 6920       015220   ADC #32      ; #bytes per P/M
5F52 85D4       015240   STA R+0
                015260 ;
5F54            015280 SHOC10
5F54 BD1C60     015300   LDA BUGP0,X
5F57 990074     015320   STA P0,Y
5F5A BDDC60     015340   LDA BUGP1,X
5F5D 990075     015360   STA P1,Y
5F60 BD9C61     015380   LDA BUGP2,X
5F63 990076     015400   STA P2,Y
5F66 BD5C62     015420   LDA BUGP3,X
5F69 990077     015440   STA P3,Y
5F6C C8         015460   INY
5F6D E8         015480   INX
5F6E E4D4       015500   CPX R+0
5F70 D0E2       015520   BNE SHOC10
5F72            015540 SHOC20
5F72 207E5F     015560   JSR WHEELS   ; spin the wheels
5F75            015570 SHOC30
5F75 60         015580   RTS
                015600 ;
                015620 ;
5F76            015640 M32TBL ;      *32 table
5F76 00204060   015660   .BYTE 0,32,64,96,128,160,192,224
5F7A 80A0C0E0
                015662 ;
                015680 ;
                015700 ;
                015720 ; DO WHEEL SPIN
                015740 ;
5F7E            015760 WHEELS
```

Moore Business Forms, Inc. sv 4128H

```
                    015740 ;
5F7E                015760 WHEELS


PAGE 41   UPN DOWN - for Atari systems
TRACK                                                    D2:TRACK3.M65

5F7E AD135B         015780   LDA SCRFLG  ; only spin while scrolling
5F81 F05E           015800   BEQ WHEE20
5F83 AD185B         015820   LDA JMPNDX  ; no spin while jumping
5F86 1059           015840   BPL WHEE20
5F88 A5DD           015860   LDA R+9     ; choose data depending on direction
5F8A 297F           015880   AND #$7F
5F8C C903           015900   CMP #3
5F8E 900E           015920   BCC WHEE05
5F90 38             015940   SEC         ; facing right
5F91 E903           015960   SBC #3
5F93 A0C1           015980   LDY #$C1
5F95 84D5           016000   STY R+1
5F97 A0A0           016020   LDY #$A0
5F99 84DD           016040   STY R+9
5F9B 4CA65F         016060   JMP WHEE10
5F9E                016080 WHEE05
5F9E A005           016100   LDY #$05    ; facing left
5FA0 84D5           016120   STY R+1
5FA2 A083           016140   LDY #$83
5FA4 84DD           016160   STY R+9
5FA6                016180 WHEE10
5FA6 0A             016200   ASL A
5FA7 0A             016220   ASL A
5FA8 85D4           016240   STA R+0
5FAA CE195B         016300   DEC WHLSEQ
5FAD AD195B         016340   LDA WHLSEQ  ; wheel sequence ctr
5FB0 2901           016360   AND #1      ; 2 positions for wheels
5FB2 05D4           016380   ORA R+0
5FB4 AA             016400   TAX
5FB5 ADA742         016420   LDA BUGY
5FB8 18             016440   CLC
5FB9 6910           016460   ADC #16
5FBB 0A             016480   ASL A
5FBC 85D7           016500   STA R+3
5FBE 7DE25F         016520   ADC WHLPOS,X ; carry clear
5FC1 A8             016540   TAY
5FC2 A5D5           016560   LDA R+1
5FC4 990074         016580   STA P0,Y
5FC7 A5DD           016600   LDA R+9
5FC9 990076         016620   STA P2,Y
5FCC 8A             016640   TXA
5FCD 4901           016660   EOR #1
5FCF 05D4           016680   ORA R+0
5FD1 AA             016700   TAX
5FD2 BDE25F         016720   LDA WHLPOS,X
5FD5 18             016740   CLC
5FD6 65D7           016760   ADC R+3
5FD8 A8             016780   TAY
5FD9 A900           016800   LDA #0
5FDB 990074         016820   STA P0,Y
5FDE 990076         016840   STA P2,Y
5FE1                016860 WHEE20
5FE1 60             016880   RTS
                    016900 ;
5FE2                016920 WHLPOS ;      Y offset for wheel marks
5FE2 1A1B1C1D       016940   .BYTE 26,27,28,29 ; level
5FE6 1B1C1D1E       016960   .BYTE 27,28,29,30 ; up
5FEA 1A1B1C1D       016980   .BYTE 26,27,28,29 ; down
```

Moore Business Forms, Inc. 8V 1472RH

```
5FE6 1B1C1D1E   016960   .BYTE 27,28,29,30 ; up
5FEA 1A1B1C1D   016980   .BYTE 26,27,28,29 ; down
```



```
                017000 ;
                017020 ;
                017040 ;
                017060 ; ADD TO PLAYER1'S SCORE
                017080 ;
5FEE            017100 ADDP1
5FEE AD0040     017120   LDA PSCOR0
5FF1 85D6       017140   STA R+2
5FF3 AD0140     017160   LDA PSCOR1
5FF6 85D7       017180   STA R+3
5FF8 AD0240     017200   LDA PSCOR2
5FFB 85D8       017220   STA R+4
5FFD A9BA       017240   LDA # <HMEM+6
5FFF 85D4       017260   STA R+0
6001 A947       017280   LDA # >HMEM+6
6003 85D5       017300   STA R+1
6005 20AE53     017320   JSR ADDSCOR
6008 A5D6       017340   LDA R+2
600A 8D0040     017360   STA PSCOR0
600D A5D7       017380   LDA R+3
600F 8D0140     017400   STA PSCOR1
6012 A5D8       017420   LDA R+4
6014 8D0240     017440   STA PSCOR2
6017 60         017460   RTS
                017480 ;
6018            017500 BUGH0 *= *+1
6019            017520 BUGH1 *= *+1
601A            017540 BUGH2 *= *+1
601B            017560 BUGH3 *= *+1
                014480   .INCLUDE #D2:PMCAR.M65
                0900     .PAGE "PM DATA"
```

Moore Business Forms, Inc. sv 1412RH

```
                        1000 ; DATA FOR PLAYER'S CAR
                        1020 ; CAR IS A PLAYER MISSILE GRAPHIC
601C                    1040 BUGP0
                        1060 ; P0    LEFT, LEVEL
601C 00010307           1080     .BYTE 0,1,3,7,7,15,59,59
6020 070F3B3B
6024 0B39697D           1100     .BYTE 11,57,105,125,117,23,27,27
6028 75171B1B
602C 584F4F4F           1120     .BYTE 88,79,79,79,10,3,1,1
6030 0A030101
6034 04040404           1140     .BYTE 4,4,4,4,4,4,0,0
6038 04040000
                        1160 ; P0   LEFT, UP
603C 307E7E38           1180     .BYTE 48,126,126,56,57,63,63,55
6040 393F3F37
6044 1B193D2D           1200     .BYTE 27,25,61,45,44,40,10,10
6048 2C280A0A
604C 02050506           1220     .BYTE 2,5,5,6,6,15,3,3
6050 060F0303
6054 07050504           1240     .BYTE 7,5,5,4,4,4,0,0
6058 04040000
                        1260 ; P0  LEFT, DOWN
605C 00000001           1280     .BYTE 0,0,0,1,3,3,7,7
6060 03030707
6064 070D0909           1300     .BYTE 7,13,9,9,9,57,59,15
6068 09393B0F
606C 347B7F7F           1320     .BYTE 52,123,127,127,27,19,17,81
6070 1B131151
6074 44444404           1340     .BYTE 68,68,68,4,4,4,0,0
6078 04040000
                        1360 ; P0   RIGHT, LEVEL
607C 00070F0F           1380     .BYTE 0,7,15,15,31,31,31,63
6080 1F1F1F3F
6084 33212121           1400     .BYTE 51,33,33,33,33,33,33,127
6088 2121217F
608C 55D4FFFD           1420     .BYTE 85,212,255,253,60,253,193,193
6090 3CFDC1C1
6094 00000000           1440     .BYTE 0,0,0,0,0,0,0,0
6098 00000000
                        1460 ; P0   RIGHT, UP
609C 0C1F1F1C           1480     .BYTE 12,31,31,28,31,31,63,63
60A0 1F1F3F3F
60A4 3F3F3F3F           1500     .BYTE 63,63,63,63,63,63,51,33
60A8 3F3F3321
60AC 21617FD5           1520     .BYTE 33,97,127,213,213,254,253,253
60B0 D5FEFDFD
60B4 FDC1C100           1540     .BYTE 253,193,193,0,0,0,0,0
60B8 00000000
                        1560 ; P0   RIGHT, DOWN
60BC 00000007           1580     .BYTE 0,0,0,7,15,15,31,31
60C0 0F0F1F1F
60C4 33212121           1600     .BYTE 51,33,33,33,33,33,33,127
60C8 2121217F
60CC 5554FFFF           1620     .BYTE 85,84,255,255,253,253,193,193
60D0 FDFDC1C1
60D4 00000000           1640     .BYTE 0,0,0,0,0,0,0,0
60D8 00000000
60DC                    1660 BUGP1
```

Moore Business Forms, Inc. sv T412RH

```
60D8 00000000
60DC            1660 BUGP1


                1680 ; P1   LEFT, LEVEL
60DC 00000001   1700      .BYTE 0,0,0,1,3,3,51,49
60E0 03033331
60E4 11063672   1720      .BYTE 17,6,54,114,154,248,236,236
60E8 9AF8ECEC
60EC EFF6F6F3   1740      .BYTE 239,246,246,243,247,109,14,14
60F0 F76D0E0E
60F4 0F0F0F0F   1760      .BYTE 15,15,15,15,15,15,7,6
60F8 0F0F0706
                1780 ; P1   LEFT, UP
60FC 00062624   1800      .BYTE 0,6,38,36,22,80,81,88
6100 16505158
6104 6C646A7A   1820      .BYTE 108,100,106,122,123,127,125,61
6108 7B7F7D3D
610C 0D0E0E0B   1840      .BYTE 13,14,14,11,11,10,13,13
6110 0B0A0D0D
6114 0D0E0E0F   1860      .BYTE 13,14,14,15,15,15,7,6
6118 0F0F0706
                1880 ; P1   LEFT, DOWN
611C 00000000   1900      .BYTE 0,0,0,0,0,1,3,3
6120 00010303
6124 01000002   1920      .BYTE 1,0,0,2,6,54,52,16
6128 06363410
612C 0B24769B   1940      .BYTE 11,36,118,155,245,236,238,238
6130 F5ECEEEE
6134 FFFFFFFF   1960      .BYTE 255,255,255,255,111,15,7,6
6138 6F0F0706
                1980 ; P1   RIGHT, LEVEL
613C 0000070F   2000      .BYTE 0,0,7,15,15,31,31,19
6140 0F1F1F13
6144 0101000E   2020      .BYTE 1,1,0,14,30,30,30,0
6148 1E1E1E00
614C 2A2B0082   2040      .BYTE 42,43,0,130,195,2,62,62
6150 C3023E3E
6154 FFFFF1F1   2060      .BYTE 255,255,241,241,241,240,96,96
6158 F1F06060
                2080 ; P1   RIGHT UP
615C 00030302   2100      .BYTE 0,3,3,2,0,7,31,31
6160 0007F1F
6164 3F3F3F1E   2120      .BYTE 63,63,63,30,12,0,12,30
6168 0C000C1E
616C 1E1E002A   2140      .BYTE 30,30,0,42,42,1,195,195
6170 2A01C3C3
6174 C33E30F1   2160      .BYTE 195,62,48,241,241,241,225,96
6178 F1F1E160
                2180 ; P1   RIGHT, DOWN
617C 00000000   2200      .BYTE 0,0,0,0,7,15,15,19
6180 070F0F13
6184 01000E1E   2220      .BYTE 1,0,14,30,30,30,30,0
6188 1E1E1E00
618C 2A2B00C1   2240      .BYTE 42,43,0,193,195,2,62,62
6190 C3023E3E
6194 FFFFFDFD   2260      .BYTE 255,255,253,253,241,240,224,96
6198 F1F0E060
619C            2280 BUGP2
                2300 ; P2   LEFT, LEVEL
619C 00E0F0F0   2320      .BYTE 0,224,240,240,248,248,248,252
```

Moore Business Forms, Inc. sv 1412RR

```
                2620    .BYTE 0,224,240,240,248,248,248,252
F8F8FC
```



```
61A4 CC848484   2340    .BYTE 204,132,132,132,132,132,132,254
61A8 848484FE
61AC AA2BFFBF   2360    .BYTE 170,43,255,191,60,191,131,131
61B0 3CBF8383
61B4 00000000   2380    .BYTE 0,0,0,0,0,0,0,0
61B8 00000000
                2400 ; P2   LEFT, UP
61BC 30F8F838   2420    .BYTE 48,248,248,56,248,248,252,252
61C0 F8F8FCFC
61C4 FCFCFCFC   2440    .BYTE 252,252,252,252,252,252,204,132
61C8 FCFCCC84
61CC 8486FEAB   2460    .BYTE 132,134,254,171,171,127,191,191
61D0 AB7FBFBF
61D4 BF838300   2480    .BYTE 191,131,131,0,0,0,0,0
61D8 00000000
                2500 ; P2   LEFT, DOWN
61DC 000000E0   2520    .BYTE 0,0,0,224,240,240,248,248
61E0 F0F0F8F8
61E4 CC848484   2540    .BYTE 204,132,132,132,132,132,132,254
61E8 848484FE
61EC AA2AFFFF   2560    .BYTE 170,42,255,255,191,191,131,131
61F0 BFBF8383
61F4 00000000   2580    .BYTE 0,0,0,0,0,0,0,0
61F8 00000000
                2600 ; P2   RIGHT, LEVEL
61FC 0080C0E0   2620    .BYTE 0,128,192,224,224,240,220,220
6200 E0F0DCDC
6204 D09C96BE   2640    .BYTE 208,156,150,190,174,232,216,216
6208 AEE8D8D8
620C 1AF2F2F2   2660    .BYTE 26,242,242,242,80,192,128,128
6210 50C08080
6214 20202020   2680    .BYTE 32,32,32,32,32,32,0,0
6218 20200000
                2700 ; P2   RIGHT, UP
621C 0C7E7E1C   2720    .BYTE 12,126,126,28,156,252,252,236
6220 9CFCFCEC
6224 D898BCB4   2740    .BYTE 216,152,188,180,52,20,80,80
6228 34145050
622C 40A0A060   2760    .BYTE 64,160,160,96,96,240,192,192
6230 60F0C0C0
6234 E0A0A020   2780    .BYTE 224,160,160,32,32,32,0,0
6238 20200000
                2800 ; P2   RIGHT, DOWN
623C 00000080   2820    .BYTE 0,0,0,128,192,192,224,224
6240 C0C0E0E0
6244 E0B09090   2840    .BYTE 224,176,144,144,144,156,220,240
6248 909CDCF0
624C 2CDEFEFE   2860    .BYTE 44,222,254,254,216,200,136,138
6250 D8C8888A
6254 22222220   2880    .BYTE 34,34,34,32,32,32,0,0
6258 20200000
625C            2900 BUGP3
                2920 ; P3   LEFT, LEVEL
625C 0000E0F0   2940    .BYTE 0,0,224,240,240,248,248,200
6260 F0F8F8C8
6264 80800070   2960    .BYTE 128,128,0,112,120,120,120,0
6268 78787800
626C 54D40041   2980    .BYTE 84,212,0,65,195,64,124,124
```

Moore Business Forms, Inc. 1412RH

```
6268 78787800
626C 54D40041    2980      .BYTE 84,212,0,65,195,64,124,124



PAGE 46   UPN DOWN - for Atari systems
PM DATA                                          D2:PMCAR.M65


6270 C3407C7C
6274 FFFF8F8F    3000      .BYTE 255,255,143,143,143,15,6,6
6278 8F0F0606
                 3020 ; P3   LEFT, UP
627C 00C0C040    3040      .BYTE 0,192,192,64,0,224,248,248
6280 00E0F8F8
6284 FCFCFC78    3060      .BYTE 252,252,252,120,48,0,48,120
6288 30003078
628C 78780054    3080      .BYTE 120,120,0,84,84,128,195,195
6290 5480C3C3
6294 C37C0C8F    3100      .BYTE 195,124,12,143,143,143,135,6
6298 8F8F8706
                 3120 ; P3   LEFT, DOWN
629C 00000000    3140      .BYTE 0,0,0,0,224,240,240,200
62A0 E0F0F0C8
62A4 80007078    3160      .BYTE 128,0,112,120,120,120,120,0
62A8 78787800
62AC 54D40083    3180      .BYTE 84,212,0,131,195,64,124,124
62B0 C3407C7C
62B4 FFFFBFBF    3200      .BYTE 255,255,191,191,143,15,7,6
62B8 8F0F0706
                 3220 ; P3   RIGHT, LEVEL
62BC 00000080    3240      .BYTE 0,0,0,128,192,192,204,140
62C0 C0C0CC8C
62C4 88606C4E    3260      .BYTE 136,96,108,78,89,31,55,55
62C8 591F3737
62CC F76F6FCF    3280      .BYTE 247,111,111,207,239,182,112,112
62D0 EFB67070
62D4 F0F0F0F0    3300      .BYTE 240,240,240,240,240,240,224,96
62D8 F0F0E060
                 3320 ; P3   RIGHT, UP
62DC 00606424    3340      .BYTE 0,96,100,36,104,10,138,26
62E0 680A8A1A
62E4 3626565E    3360      .BYTE 54,38,86,94,222,254,190,188
62E8 DEFEBEBC
62EC B07070D0    3380      .BYTE 176,112,112,208,208,80,176,176
62F0 D050B0B0
62F4 B07070F0    3400      .BYTE 176,112,112,240,240,240,224,96
62F8 F0F0E060
                 3420 ; P3   RIGHT, DOWN
62FC 00000000    3440      .BYTE 0,0,0,0,0,128,192,192
6300 0080C0C0
6304 80000040    3460      .BYTE 128,0,0,64,96,108,44,8
6308 606C2C08
630C D0246ED9    3480      .BYTE 208,36,110,217,175,55,119,119
6310 AF377777
6314 FFFFFFFF    3500      .BYTE 255,255,255,255,246,240,224,96
6318 F6F0E060
                 3600 ;    END OF PLAYER P/M DATA
                 014500    .INCLUDE #D2:MOVE.M65
                 1000      .PAGE "MOVE"
```

```
                    1020 ; UP'N DOWN
                    1040 ; MOVE.M65
                    1060 ;
                    1080 ; TABLES FOR OBJECT ENTRIES
                    1100 ;
        =0001       1120 OBJMAX = 1      ; max # of objects on screen
        =000E       1140 YCRSH = 14      ; Y tolerance on collision detect
        =000F       1160 XCRSH = 15
    631C            1180 REDRWO *= *+1   ;redraw obj0 flag
    631D            1200 TYPE *= *+9
    6326            1220 XPOS *= *+9
    632F            1240 YPOS *= *+9
    6338            1260 DIR *= *+9
    6341            1280 SPEED *= *+9
    634A            1300 SPDCTR *= *+9
    6353            1320 MODE *= *+9
    635C            1340 PATH *= *+9
    6365            1360 LASTINC *= *+9
    636E            1380 LASTNZR *= *+9
    6377            1400 OVRLAY *= *+9
                    1420 ;
                    1440 ; ROUTINE TO MOVE ALL OBJECTS IN
                    1460 ; THE OBJECT TABLE
                    1480 ;
    6380            1500 MOVE
    6380 A200       1520     LDX #OBJMAX-1 ; max # objects
    6382            1540 MOVE10
    6382 BD1D63     1560     LDA TYPE,X  ; active entry?
    6385 1004       1580     BPL MOVE16  ; yes - branch
    6387            1600 MOVE15
    6387 CA         1620     DEX
    6388 10F8       1640     BPL MOVE10
    638A 60         1660     RTS
    638B            1680 MOVE16
    638B BD4163     1700     LDA SPEED,X ; time to move?
    638E 1005       1720     BPL MOVE20
    6390 49FF       1740     EOR #$FF
    6392 18         1760     CLC
    6393 6901       1780     ADC #1
    6395            1800 MOVE20
    6395 18         1820     CLC
    6396 7D4A63     1840     ADC SPDCTR,X
    6399 9D4A63     1860     STA SPDCTR,X
    639C 90E9       1880     BCC MOVE15  ; no move - branch
                    1900 ;
    639E A900       1920     LDA #0
    63A0 8D1C63     1940     STA REDRWO
    63A3 E000       1960     CPX #0
    63A5 F00F       1980     BEQ MOVE25
    63A7 202A64     2000     JSR OVRLAP
    63AA 900A       2020     BCC MOVE25
    63AC A200       2040     LDX #0
    63AE 200E5A     2060     JSR ERASE
    63B1 A201       2080     LDX #1
    63B3 8E1C63     2100     STX REDRWO
    63B6            2120 MOVE25
    63B6 200E5A     2140     JSR ERASE   ; erase this truck
    63B9 BC2F63     2160     LDY YPOS,X  ; move it
```


Moore Business Forms, Inc. sv 14128H

```
63B6 200E5A     2140     JSR ERASE    ; erase this truck
63B9 BC2F63     2160     LDY YPOS,X   ; move it
```



```
63BC BD4163     2180     LDA SPEED,X
63BF 1004       2200     BPL MOVE30
63C1 C8         2220     INY
63C2 4CC663     2240     JMP MOVE40
63C5            2260 MOVE30
63C5 88         2280     DEY
63C6            2300 MOVE40
63C6 C060       2320     CPY #96      ; get rid of trucks that move off screen
63C8 B04D       2340     BCS MOVE50
63CA 84D5       2360     STY R+1
63CC BD2663     2380     LDA XPOS,X
63CF C9A0       2400     CMP #160
63D1 B044       2420     BCS MOVE50
63D3 85D4       2440     STA R+0
63D5 BD3B63     2460     LDA DIR,X
63D8 85D6       2480     STA R+2
63DA BD5C63     2500     LDA PATH,X
63DD 85DC       2520     STA R+8
63DF BD6563     2540     LDA LASTINC,X
63E2 85DE       2560     STA R+10
63E4 BD6E63     2580     LDA LASTNZR,X
63E7 85DF       2600     STA R+11
63E9 BD4163     2620     LDA SPEED,X
63EC 85E0       2640     STA R+12
63EE 201A5B     2660     JSR TRACK
                2680 ;
                2700 ; update new parms
                2720 ;
63F1 A5D4       2740     LDA R+0
63F3 9D2663     2760     STA XPOS,X
63F6 A5D5       2780     LDA R+1
63F8 9D2F63     2800     STA YPOS,X
63FB A5D6       2820     LDA R+2
63FD 9D3B63     2840     STA DIR,X
6400 A5DC       2860     LDA R+8
6402 9D5C63     2880     STA PATH,X
6405 A5DE       2900     LDA R+10
6407 9D6563     2920     STA LASTINC,X
640A A5DF       2940     LDA R+11
640C 9D6E63     2960     STA LASTNZR,X
                2980 ;
640F 20A859     3000     JSR DRAW     ; draw at new position
6412 B003       3020     BCS MOVE50   ; draw not successful - branch
6414 4C8763     3040     JMP MOVE15
6417            3060 MOVE50
6417 A9FF       3080     LDA #-1      ; cancel object
6419 9D1D63     3100     STA TYPE,X
641C 4C8763     3120     JMP MOVE15
                3140 ;
                3160 ; INIT THE OBJECT TABLES
                3180 ;
641F            3200 IOBJTB
641F A9FF       3220     LDA #-1      ; inactive flag value
6421 A200       3240     LDX #OBJMAX-1 ; # entries
6423            3260 IOBJ10
6423 9D1D63     3280     STA TYPE,X
6426 CA         3300     DEX
6427 10FA       3320     BPL IOBJ10
```

Moore Business Forms, Inc. sv 1412RH

```
                        3340 ;
6429 60                 3360    RTS
                        3380 ;
                        3400 ; See if trucks overlap
                        3420 ; Yreg = # of first overlap
                        3440 ;      = $FF if no overlap
                        3442 ; Uses R+18
                        3444 ;
642A                    3460 OVRLAP
642A 86E6               3462    STX R+18
642C A000               3550    LDY #OBJMAX-1
                        3554 ;
642E                    3560 OVRL05  6323
642E B91D63             3580    LDA TYPE,Y  ; check for entries
6431 3029               3600    BMI OVRL40  ; can't overlap if not drawn
6433 C4E6               3602    CPY R+18    ; can't ovrlap self
6435 F025               3604    BEQ OVRL40
                        3620 ;
6437 B92663             3640    LDA XPOS,Y
643A 38                 3660    SEC
643B FD2663             3680    SBC XPOS,X
643E 1005               3700    BPL OVRL10
6440 49FF               3720    EOR #$FF
6442 18                 3740    CLC
6443 6901               3760    ADC #1
6445                    3780 OVRL10
6445 C910               3800    CMP #16
6447 B013               3820    BCS OVRL40  ; no ovrlap
                        3840 ;
6449 B92F63             3860    LDA YPOS,Y
644C 38                 3880    SEC
644D FD2663             3900    SBC XPOS,X
6450 1005               3920    BPL OVRL20
6452 49FF               3940    EOR #$FF
6454 18                 3960    CLC
6455 6901               3980    ADC #1
6457                    4000 OVRL20
6457 C910               4020    CMP #16
6459 B001               4040    BCS OVRL40
                        4060 ;
645B                    4070 OVRL30 ;       found an overlap
645B 60                 4100    RTS
645C                    4120 OVRL40
645C 88                 4122    DEY
645D 10CF               4124    BPL OVRL05
645F A0FF               4140    LDY #$FF
6461 60                 4160    RTS
                        4180 ;
                        4200 ;
                        4220 ; INIT AN ENEMY TRUCK
                        4240 ; Enter:
                        4260 ;      TPNDX = # of type to init (0-5)
                        4280 ;      TPROB = (ROUND-1)*6
                        4300 ;      R+0,1,7,8,9,10
                        4320 ; USES:
                        4340 ;
6462                    4360 TINIT
                        4380 ; Check probablities to see if to init truck
```

LDA #$C8
AND
BNE #24
LDA
STA
LDA #$3F
AND
BNE

Moore Business Forms, Inc. ev T412RH

```
6462                4360 TINIT
                    4380 ; Check probablities to see if to init truck
```



```
                    4400 ;
6462 AC8865         4420     LDY TPNDX   ; get index to truck parameters
6465 A206           4440     LDX #NTYPES
6467                4460 TINI10
6467 B95E65         4480     LDA TPROB,Y ; get probability of init
646A F005           4500     BEQ TINI15
646C CD0AD2         4520     CMP RANDOM  ; init this type?
646F B005           4540     BCS TINI20  ; yes - branch
6471                4560 TINI15
6471 C8             4580     INY
6472 CA             4600     DEX
6473 D0F2           4620     BNE TINI10
6475 60             4640     RTS         ; no inits this time
                    4660 ;
                    4680 ; Find an open slot
                    4700 ;
6476                4720 TINI20
6476 A200           4740     LDX #OBJMAX-1
                    4760 ;;  LDX #1      ;;;;DEBUG
6478                4780 TINI30
6478 BD1D63         4800     LDA TYPE,X
647B 3004           4820     BMI TINI40
647D CA             4840     DEX
647E 10F8           4860     BPL TINI30
6480 60             4880     RTS         ; no open slots
                    4900 ;
                    4920 ; Only init if VW overtakes/overtaken
                    4940 ;
6481                4960 TINI40
6481 98             4980     TYA
6482 48             5000     PHA
6483 8A             5020     TXA
6484 48             5040     PHA
                    5060 ;
                    5080 ; Check speeds to see who overtakes
                    5100 ;
6485 AD0AD2         5120     LDA RANDOM  ; init spd from tables
6488 397C65         5140     AND TMAXSP,Y
648B 18             5160     CLC
648C 798265         5180     ADC TMINSP,Y
648F 85E1           5200     STA R+13
                    5220 ;
6491 A5E1           5240     LDA R+13    ; check signs
6493 4D0A40         5260     EOR SCRSPD
6496 301B           5280     BMI TIN553  ; opposite dir - branch
6498 A5E1           5300     LDA R+13    ; check magnitudes
649A 1005           5320     BPL TINI50
649C 49FF           5340     EOR #$FF
649E 18             5360     CLC
649F 6901           5380     ADC #1
64A1                5400 TINI50
64A1 85DC           5420     STA R+8
64A3 AD0A40         5440     LDA SCRSPD
64A6 1005           5460     BPL TINI52
64A8 49FF           5480     EOR #$FF
64AA 18             5500     CLC
64AB 6901           5520     ADC #1
64AD                5540 TINI52
```

Moore Business Forms, Inc. sv 1412RH

```
64AB 6901          5520      ADC #1
64AD               5540 TINI52


PAGE 51    UPN DOWN - for Atari systems
MOVE                                                    D2:MOVE.M65


64AD C5DC          5560      CMP R+8
64AF 900C          5580      BCC TIN552  ; VW is slower - truck will overtake
64B1 F014          5600      BEQ TINI54  ; same speed
64B3               5620 TIN553
64B3 AD0A40        5640      LDA SCRSPD  ; VW is faster
64B6 300A          5660      BMI TIN550
64B8               5680 TIN551
64B8 A901          5700      LDA #1      ; init at top
64BA 4CCF64        5720      JMP TINI56
64BD               5740 TIN552
64BD AD0A40        5760      LDA SCRSPD
64C0 30F6          5780      BMI TIN551
64C2               5800 TIN550
64C2 A95F          5820      LDA #95     ;  "  bottom
64C4 4CCF64        5840      JMP TINI56
64C7               5860 TINI54
64C7 AD0AD2        5880      LDA RANDOM  ; select top/bot randomly
64CA 10F6          5900      BPL TIN550
64CC 4CB864        5920      JMP TIN551
64CF               5940 TINI56
64CF 85DB          5960      STA R+7     ; save
64D1 18            5980      CLC
64D2 690B          6000      ADC #YTRKPT ; find road at track point
64D4 6D0840        6020      ADC DSCROL
64D7 AA            6040      TAX
64D8 BD3C40        6060      LDA DSPTBL,X ; set ptr to line
64DB 85D4          6080      STA R+0
64DD BD1B41        6100      LDA DSPTBH,X
64E0 85D5          6120      STA R+1
                   6140 ; Put truck on road
                   6160 ; Search for road
                   6180 ;
64E2 A027          6200      LDY #39     ; 40 bytes/line
64E4 A200          6220      LDX #0
64E6               6240 TINI60
64E6 B1D4          6260      LDA (R+0),Y
64E8 C9FF          6280      CMP #$FF    ; look for road on this line
64EA D029          6300      BNE TINI70
64EC 88            6320      DEY         ; must get at least 2 road bytes to be
                                         sure
64ED D1D4          6340      CMP (R+0),Y
64EF D024          6360      BNE TINI70
64F1 88            6380      DEY         ; now find next non-road byte
64F2 D1D4          6400      CMP (R+0),Y
64F4 D005          6420      BNE TINI62
64F6 88            6440      DEY         ; 4 road bytes is too many - can't be road
64F7 D1D4          6460      CMP (R+0),Y
64F9 F01A          6480      BEQ TINI70
64FB               6500 TINI62
64FB A903          6520      LDA #3      ; find left pixel of road
64FD 85DE          6540      STA R+10
64FF B1D4          6560      LDA (R+0),Y
6501               6580 TINI64
6501 6A            6600      ROR A
6502 9008          6620      BCC TINI66
6504 6A            6640      ROR A
6505 9005          6660      BCC TINI66
6507 C6DE          6680      DEC R+10
```

Moore Business Forms, Inc. xv 1412RH

```
6505 9005       6660     BCC TINI66
6507 C6DE       6680     DEC R+10


PAGE 52   UPN DOWN - for Atari systems
MOVE                                          D2:MOVE.M65

6509 4C0165     6700     JMP TINI64
650C            6720 TINI66
650C 98         6740     TYA          ;convert to Xpos
650D 0A         6760     ASL A
650E 0A         6780     ASL A
650F 18         6800     CLC
6510 65DE       6820     ADC R+10
6512 95D6       6840     STA R+2,X    ; save
6514 E8         6860     INX
6515            6880 TINI70
6515 88         6900     DEY
6516 10CE       6920     BPL TINI60
6518 E000       6940     CPX #0       ; find any roads?
651A F00B       6960     BEQ TINI74   ; no - branch
651C 86DA       6980     STX R+6      ;save # roads found
                7000 ;
651E ADOAD2     7020     LDA RANDOM   ; figure out which Xpos to use
6521 2903       7040     AND #$03     ; max #roads found=4
                7060 ;
6523 C5DA       7080     CMP R+6      ; valid road?
6525 9005       7100     BCC TINI80   ; yes - branch
6527            7120 TINI74
6527 68         7140     PLA          ; can't init
6528 AA         7160     TAX
6529 68         7180     PLA
652A A8         7200     TAY
652B 60         7220     RTS
                7240 ;
652C            7260 TINI80
652C AA         7280     TAX
652D B5D6       7300     LDA R+2,X
652F 85DA       7320     STA R+6      ; save initial Xpos
                7340 ;
6531 68         7360     PLA
6532 AA         7380     TAX
6533 68         7400     PLA
6534 A8         7420     TAY
                7440 ;
                7460 ;
                7480 ; Init truck's parameters
                7500 ;
6535 A5DA       7520     LDA R+6  631B
6537 9D2663     7540     STA XPOS,X
653A A5DB       7560     LDA R+7  6324
653C 9D2F63     7580     STA YPOS,X
653F A5E1       7600     LDA R+13
6541 9D4163     7620     STA SPEED,X
6544 98         7640     TYA      6312
6545 9D1D63     7660     STA TYPE,X
                7680 ;
6548 202A64     7682     JSR OVRLAP   ; see if overlaps
654B            7683 TINI90
654B C0FF       7684     CPY #$FF
654D F007       7686     BEQ TIN100   ; no overlap
654F B97763     7688     LDA OVRLAY,Y ; find end of chain
6552 A8         7690     TAY
6553 4C4B65     7692     JMP TINI90
6556            7694 TINI00
```

Moore Business Forms, Inc. sv 1412RH

```
                        JMP TINI90
                7694 TIN100
```



```
6556 98         7700    TYA
6557 9D7763     7720    STA OVRLAY,X
655A 20A859     7740    JSR DRAW    ; initial draw
655D 60         7760    RTS
                7780 ;
                7800 ; TABLE OF PROBABILITIES TO INIT TRUCKS
                7820 ; PROB is x/255 per call to TINIT
655E            7840 TPROB
655E 01000000   7860    .BYTE 1,0,0,0,0,0 ; ROUND 1
6562 0000
6564 05050500   7880    .BYTE 5,5,5,0,0,0 ; ROUND 2
6568 0000
656A 00050505   7900    .BYTE 0,5,5,5,0,0 ;     3
656E 0000
6570 00000505   7920    .BYTE 0,0,5,5,5,0 ;     4
6574 0500
6576 00000005   7940    .BYTE 0,0,0,5,5,5 ;     5
657A 0505
                7960 ; Initial speeds for trucks
                7980 ;
657C            8000 TMAXSP ;       truck max speed (ANDed)
657C 07070707   8020    .BYTE 7,7,7,7,7,7
6580 0707
6582            8040 TMINSP ;       truck min speed (ADDed)
6582 01010101   8060    .BYTE 1,1,1,1,1,1
6586 0101
                8080 ;
6588            8100 TPNDX *= *+1
   =0006        8120 NTYPES = 6     ; number of truck types
                8140 ;
                8160 ; COLLISION ROUTINE
                8180 ; CHECKS FOR COLLISIONS BETWEEN
                8200 ; PLAYER'S VW AND ENEMY TRUCKS
                8220 ;
                8240 ; Returns:
                8260 ;      carry set and Xreg has index
                8280 ;      of truck collided with ,
                8300 ; or
                8320 ;      carry clear if no collision
                8340 ;
6589            8360 COLISN
6589 A200       8380    LDX #OBJMAX-1 ; check all OBJTBL entries
658B            8400 COLI10
658B BD1D63     8420    LDA TYPE,X  ; find active entry
658E 3024       8440    BMI COLI40
6590 BD2F63     8460    LDA YPOS,X  ; check Y
6593 38         8480    SEC
6594 EDA742     8500    SBC BUGY
6597 B005       8520    BCS COLI20  ; find absolute difference
6599 49FF       8540    EOR #$FF
659B 18         8560    CLC
659C 6901       8580    ADC #1
659E            8600 COLI20
659E C90E       8620    CMP #YCRSH  ; Y tolerance on collision
65A0 B012       8640    BCS COLI40  ; no hit - branch
                8660 ;
65A2 BD2663     8680    LDA XPOS,X  ; check X
65A5 38         8700    SEC
```

Moore Business Forms, Inc. sv 1412RH

```
65A6 EDA642     8720     SBC BUGX
65A9 B005       8740     BCS COLI30
65AB 49FF       8760     EOR #$FF
65AD 18         8780     CLC
65AE 6901       8800     ADC #1
65B0            8820 COLI30
65B0 C90F       8840     CMP #XCRSH  ; X tolerance
65B2 9005       8860     BCC COLI50  ; found a collision - branch
65B4            8880 COLI40
65B4 CA         8900     DEX         ; next object
65B5 10D4       8920     BPL COLI10
65B7 18         8940     CLC         ; no collisions
65B8 60         8960     RTS
65B9            8980 COLI50
65B9 38         9000     SEC         ; return with collided truck index in
                                     Xreg.
65BA 60         9020     RTS
                9040 ;
                9060 ;
                9080 ; INIT PARAMS FOR NEXT ROUND
                9100 ;
65BB            9120 NEWRND
65BB AD0340     9140     LDA ROUND
65BE EE0340     9160     INC ROUND   ; inc round ctr
65C1 0A         9180     ASL A       ; TPNDX=ROUND*6
65C2 85D4       9200     STA R+0
65C4 0A         9220     ASL A
65C5 18         9240     CLC
65C6 65D4       9260     ADC R+0
65C8 8D8865     9280     STA TPNDX
65CB 60         9300     RTS
                014520   .INCLUDE #D2:SOUND.M65
                0900     .PAGE "SOUND"
```

Moore Business Forms, Inc. sv 1412RH

```
                    1000 ; SOUND.M65
                    1020 ;
                    1040 ; UPN DOWN SOUND ROUTINES AND DATA
                    1060 ;
                    1080 ;
                    1100 ; THE FOLLOWING ROUTINES INIT THE
                    1120 ; VARIOUS SOUNDS
                    1140 ;
    65CC            1160 VWJUMP
    65CC A928       1180     LDA #$28
    65CE 8D6E67     1200     STA AUDV3
    65D1 A93F       1220     LDA # <SSTR1
    65D3 8D5667     1240     STA VOICE3
    65D6 A966       1260     LDA # >SSTR1
    65D8 8D5767     1280     STA VOICE3+1
    65DB 60         1300     RTS
                    1320 ;
    65DC            1340 TRKDIE
    65DC A928       1360     LDA #$28
    65DE 8D7067     1380     STA AUDV4
    65E1 A91B       1400     LDA # <SSTR2
    65E3 8D5867     1420     STA VOICE4
    65E6 A966       1440     LDA # >SSTR2
    65E8 8D5967     1460     STA VOICE4+1
    65EB 60         1480     RTS
                    1500 ;
    65EC            1520 VWDIE
    65EC A9A8       1540     LDA #$A8
    65EE 8D6A67     1560     STA AUDV1
    65F1 A963       1580     LDA # <SSTR3
    65F3 8D5267     1600     STA VOICE1
    65F6 A966       1620     LDA # >SSTR3
    65F8 8D5367     1640     STA VOICE1+1
    65FB A9A8       1660     LDA #$A8
    65FD 8D6C67     1680     STA AUDV2
    6600 A995       1700     LDA # <SSTR32
    6602 8D5467     1720     STA VOICE2
    6605 A966       1740     LDA # >SSTR32
    6607 8D5567     1760     STA VOICE2+1
    660A 60         1780     RTS
                    1800 ;
    660B            1820 FLAG
    660B A9A8       1840     LDA #$A8
    660D 8D7067     1860     STA AUDV4
    6610 A9C7       1880     LDA # <SSTR4
    6612 8D5867     1900     STA VOICE4
    6615 A966       1920     LDA # >SSTR4
    6617 8D5967     1940     STA VOICE4+1
    661A 60         1960     RTS
                    1980 ;
                    2000 ;
                    2020 ;
                    2040 ;
                    2060 ;
                    2080 ; TRUCK DIE SOUND
    661B            2100 SSTR2
    661B 09010801   2120     .BYTE $09,1,$08,1,$07,1,$06,1
    661F 07010601
```

```
661B 09010801   2120      .BYTE $09,1,$08,1,$07,1,$06,1
661F 07010601

PAGE 56   UPN DOWN - for Atari systems
SOUND                                          D2:SOUND.M65

6623 05010401   2140      .BYTE $05,1,$04,1,$03,1,$02,1
6627 03010201
662B 0001       2160      .BYTE 0,1
662D 09010801   2180      .BYTE $09,1,$08,1,$07,1,$06,1
6631 07010601
6635 05010401   2200      .BYTE $05,1,$04,1,$03,1,$02,1
6639 03010201
663D 0000       2220      .BYTE 0,0
                2240 ;
                2260 ; VW JUMP SOUND
663F            2280 SSTR1
663F 12011001   2300      .BYTE $12,1,$10,1,$0F,1,$0E,1
6643 0F010E01
6647 0D010C01   2320      .BYTE $0D,1,$0C,1,$0B,1,$0A,1
664B 0B010A01
664F 09010801   2340      .BYTE $09,1,$08,1,$07,1,$06,1
6653 07010601
6657 05010401   2360      .BYTE $05,1,$04,1,$03,1,$02,1
665B 03010201
665F 01000000   2380      .BYTE 1,0,0,0
                2400 ;
                2420 ; VW DIE SOUND
6663            2440 SSTR3
6663 3C040004   2460      .BYTE 60,4,0,4,60,4,0,4;47,4,0,4
6667 3C040004
666B 2F040004
666F 28040004   2480      .BYTE 40,4,0,4;33,4,0,4,35,4,0,4
6673 21040004
6677 23040004
667B 28040004   2500      .BYTE 40,4,0,4,47,4,0,4,60,4,0,4
667F 2F040004
6683 3C040004
6687 3C040004   2520      .BYTE 60,4,0,4,68,4,0,4,60,4,0,4
668B 44040004
668F 3C040004
6693 0000       2540      .BYTE 0,0
6695            2560 SSTR32
6695 60050003   2580      .BYTE 96,5,0,3,96,5,0,3,81,5,0,3
6699 60050003
669D 51050003
66A1 48050003   2600      .BYTE 72,5,0,3,69,5,0,3,72,5,0,3
66A5 45050003
66A9 48050003
66AD 51050003   2620      .BYTE 81,5,0,3,96,5,0,3,123,5,0,3
66B1 60050003
66B5 7B050003
66B9 7B050003   2640      .BYTE 123,5,0,3,123,5,0,3,123,5,0,3
66BD 7B050003
66C1 7B050003
66C5 0000       2660      .BYTE 0,0
                2680 ;
                2700 ; FLAG SOUND
66C7            2720 SSTR4
66C7 30022C02   2740      .BYTE $30,2,$2C,2,$28,2,$24,2,$20,2,$1C,2
66CB 28022402
66CF 20021C02
66D3 18021402   2760      .BYTE $18,2,$14,2,$10,2,$08,2
66D7 10020802
```

```
66D3 18021402   2760      .BYTE $18,2,$14,2,$10,2,$08,2
66D7 10020802
```



```
66DB 0000       2780      .BYTE 0,0
                2800 ;
                2820 ;
                2840 ; ROUTINE TO PLAY SOUND STRINGS
                2860 ;
                2880 ; THERE ARE 4 VOICES AVAILABLE
                2900 ; UNLESS TWO ARE CONNECTED VIA
                2920 ; AUDCTL.
                2940 ;
                2960 ; A SOUND STRING HAS THE FORM:
                2980 ;     freq, interval, freq, interval, ... 0,0
                3000 ; INTERVALS ARE 1/60th SECONDS.
                3020 ; A FREQ OF 0 WILL CAUSE A REST
                3040 ; FOR THAT INTERVAL.
                3060 ; STRINGS MUST BE TERMINATED BY 0,0.
                3080 ;
                3100 ; TO START A STRING:
                3120 ;      LOAD AUDV1-4 WITH AUDC1-4 VALUE.
                3140 ;      SET VOICE1-4 TO POINT TO THE STRING.
                3160 ;      SINCE A NON-ZERO IN THE HIGH
                3180 ;      BYTE WILL TRIGGER PLAYING LOAD THE
                3200 ;      LOW BYTE FIRST (NOTE: STRINGS
                3220 ;      CANNOT BE ON ZERO PAGE).
                3240 ;
                3260 ; EXAMPLE:
                3280 ;      LDA #$A8       ; for pure tones
                3300 ;      STA AUDV1      ; for voice 1
                3320 ;      LDA #<SSTRING  ; low byte
                3340 ;      STA VOICE1
                3360 ;      LDA #>SSTRING  ; high byte
                3380 ;      STA VOICE1+1   ; init playing
                3400 ;
                3420 ;
    66DD        3440 SFX
 66DD A200      3460       LDX #0
 66DF BD5367    3480 LP405 LDA VOICE1+1,X ; check for new sound string pointer
 66E2 F010      3500       BEQ BR463   ; if zero then voice is not active
 66E4 9D5B67    3520       STA VPTR1+1,X ; else get new pointer
 66E7 BD5267    3540       LDA VOICE1,X
 66EA 9D5A67    3560       STA VPTR1,X
 66ED A900      3580       LDA #0
 66EF 9D5367    3600       STA VOICE1+1,X
 66F2 F014      3620       BEQ BR420   ; begin playing
 66F4 BD5B67    3640 BR463 LDA VPTR1+1,X ; get high byte of voice pointer
 66F7 F008      3660       BEQ BR419   ; 0 means voice not active
 66F9 BD6267    3680       LDA VTIMR1,X ; check voice timer
 66FC F00A      3700       BEQ BR420   ; if zero then voice is available
 66FE DE6267    3720 BR461 DEC VTIMR1,X ; else decrement the timer
 6701 E8        3740 BR419 INX         ; increment index for next voice
 6702 E8        3760       INX
 6703 E008      3780       CPX #8      ; do for four voices
 6705 D0D8      3800       BNE LP405
 6707 60        3820       RTS
                3840 ;
                3860 ;  Set pointer to sound string
                3880 ;
 6708 BD5A67    3900 BR420 LDA VPTR1,X
 670B 85D4      3920       STA R+0
```

Moore Business Forms, Inc. ev 14128H

```
                        3840 ;
                        3860 ;  Set pointer to sound string
                        3880 ;
6708 BD5A67             3900 BR420 LDA VPTR1,X
670B 85D4               3920       STA R+0
```



```
670D BD5B67             3940       LDA VPTR1+1,X
6710 85D5               3960       STA R+1
                        3980 ;
                        4000 ;  Play next note
                        4020 ;
6712 A000               4040       LDY #0
6714 B1D4               4060       LDA (R+0),Y ; get new frequency
6716 D003               4080       BNE BR457   ; 0 means a rest
6718 98                 4100       TYA
6719 F00B               4120       BEQ BR458   ; unconditional jump
671B 9D00D2             4140 BR457 STA AUDF1,X ; play new freq.
671E BD6B67             4160       LDA AUDV1+1,X ; AUDV1+1 is flag for a previous rest
6721 D009               4180       BNE BR459   ; ...0 means currently resting
6723 BD6A67             4200       LDA AUDV1,X
6726 9D01D2             4220 BR458 STA AUDC1,X ; restore AUDC1
6729 9D6B67             4240       STA AUDV1+1,X ; unflag
672C C8                 4260 BR459 INY
672D B1D4               4280       LDA (R+0),Y ; get new timer interval
672F D00B               4300       BNE BR421   ; 0 means end of string
6731 9D01D2             4320       STA AUDC1,X ; stop AUDCx
6734 9D5B67             4340       STA VPTR1+1,X ; free voice
6737 9D6B67             4360       STA AUDV1+1,X ; indicate rest
673A F0C5               4380       BEQ BR419   ; unconditional jump for next voice
673C 9D6267             4400 BR421 STA VTIMR1,X ; set timer
                        4420 ;
                        4440 ;  Increment the 2-byte voice pointer
                        4460 ;
673F FE5A67             4480       INC VPTR1,X
6742 D003               4500       BNE BR423
6744 FE5B67             4520       INC VPTR1+1,X
6747 FE5A67             4540 BR423 INC VPTR1,X
674A D003               4560       BNE BR424
674C FE5B67             4580       INC VPTR1+1,X
674F 18                 4600 BR424 CLC
6750 90AC               4620       BCC BR461
                        4640 ;
                        4660 ;
6752                    4680 VOICE1 *= *+8
675A                    4700 VPTR1 *= *+8
6762                    4720 VTIMR1 *= *+8
676A                    4740 AUDV1 *= *+8
    =D201               4760 AUDC1 = $D201
    =D200               4780 AUDF1 = $D200
    =6754               4800 VOICE2 = VOICE1+2
    =6756               4820 VOICE3 = VOICE2+2
    =6758               4840 VOICE4 = VOICE3+2
    =676C               4860 AUDV2 = AUDV1+2
    =676E               4880 AUDV3 = AUDV2+2
    =6770               4900 AUDV4 = AUDV3+2
                        4920 ;
                        4940 ;
    =D208               4960 AUDCTL = $D208
    =D20F               4980 SKCTL = $D20F
    =0232               5000 SSKCTL = $0232
                        5020 ;
                        5040 ; INIT SOUND
                        5060 ;
6772                    5080 INIT
```

Moore Business Forms, Inc. sv 1412RH

```
                    5060 ;
6772                5080 ISFX



* SOUND                                                D2:SOUND.M65


                    5100 ;
6772 A207           5120     LDX #7
6774 A900           5140     LDA #0
6776                5160 ISFX10
6776 9D5267         5180     STA VOICE1,X
6779 9D6267         5200     STA VTIMR1,X
677C 9D5A67         5220     STA VPTR1,X
677F 9D6A67         5240     STA AUDV1,X
6782 CA             5260     DEX
6783 10F1           5280     BPL ISFX10
                    5300 ;
6785 A900           5320     LDA #0
6787 8D08D2         5340     STA AUDCTL
678A A903           5360     LDA #3
678C 8D0FD2         5380     STA SKCTL
678F 8D3202         5400     STA SSKCTL
6792 60             5420     RTS
                    014540   .INCLUDE #D2:SCROLL.M65
                    1000     .PAGE "SCROLL"
```

Moore Business Forms, Inc. 14128M

```
                        1020 ;
                        1040 ; SCROLL.M65
                        1060 ; SCROLL ROUTINE
                        1080 ;
6793                    1100 SCROLL
6793 A900               1120     LDA #0
6795 8D135B             1140     STA SCRFLG  ; flag no scroll
6798 AD0A40             1160     LDA SCRSPD  ; get scroll speed
679B 1005               1180     BPL SCR008  ; make positive
679D 49FF               1200     EOR #$FF
679F 18                 1220     CLC
67A0 6901               1240     ADC #1
67A2                    1260 SCR008
67A2 18                 1280     CLC
67A3 6D0940             1300     ADC SCRCTR
67A6 8D0940             1320     STA SCRCTR
67A9 9050               1340     BCC SCR030  ; if no carry , no scroll this time
67AB AD0A40             1360     LDA SCRSPD  ; flag scroll
67AE 8D135B             1380     STA SCRFLG
                        1400 ;
                        1420 ; adjust Y of objects on screen
                        1440 ;
67B1 A208               1460     LDX #X OBJMAX-1
67B3                    1480 SCR022
67B3 BD1D63             1500     LDA TYPE,X  ; check object
67B6 3020               1520     BMI SCR244  ; branch if not active
67B8 BC2F63             1540     LDY YPOS,X
67BB AD0A40             1560     LDA SCRSPD  ; add or subtract?
67BE 1004               1580     BPL SCR023
67C0 88                 1600     DEY
67C1 4CC567             1620     JMP SCR024
67C4                    1640 SCR023
67C4 C8                 1660     INY
67C5                    1680 SCR024
67C5 C060               1700     CPY #96
67C7 900B               1720     BCC SCR242
67C9 200E5A             1740     JSR ERASE   ; off screen
67CC A9FF               1760     LDA #-1
67CE 9D1D63             1780     STA TYPE,X  ; cancel
67D1 4CD867             1800     JMP SCR244
67D4                    1820 SCR242
67D4 98                 1840     TYA
67D5 9D2F63             1860     STA YPOS,X
67D8                    1880 SCR244
67D8 CA                 1900     DEX
67D9 10D8               1920     BPL SCR022
                        1940 ;
67DB                    1960 SCR233
67DB ADOBD4             1980     LDA VCOUNT  ; sync to screen
67DE C964               2000     CMP #100
67E0 90F9               2020     BCC SCR233
67E2 AD0A40             2040     LDA SCRSPD
67E5 3006               2060     BMI SCR025  68D5
67E7 20FC67             2080     JSR UPSCRN  ; scroll up
67EA 4CF067             2100     JMP SCR027
67ED                    2120 SCR025
67ED 204468             2140     JSR DNSCRN  ; scroll down
                        2160 ;
```

```
67ED 204468     2140     JSR DNSCRN  ; scroll down
                2160 ;
```



```
                2180 ;
                2200 ; add to score
                2220 ;
67F0            2240 SCR027
67F0 A902       2260     LDA #2
67F2 85D9       2280     STA R+5
67F4 A900       2300     LDA #0
67F6 85DA       2320     STA R+6
67F8 20EE5F     2340     JSR ADDP1
                2360 ;
67FB            2380 SCR030
67FB 60         2400     RTS
                2420 ;
                2440 ;
                2460 ;
                2480 ; Scrolling is accomplished by writing the new
                2500 ; line to be scrolled in over the
                2520 ; line about to be scrolled out and changing
                2540 ; the display list.
                2560 ;
                2580 ;
                2600 ; SCROLL THE DISPLAY WINDOW UP
                2620 ;
67FC            2640 UPSCRN
67FC            2652 UPSCR1
67FC CE0B40     2660     DEC SCROLF  ; dec pointer to 1st playfield line, fine
67FF 1024       2680     BPL UPSC10  ; check underflow
6801 A90F       2700     LDA #15     ; reset fine
6803 8D0B40     2720     STA SCROLF
6806 A5CA       2740     LDA SCROLC  ; dec coarse
6808 38         2760     SEC
6809 E90A       2780     SBC #10     ; 10 chars/row
680B 85CA       2800     STA SCROLC
680D A5CB       2820     LDA SCROLC+1
680F E900       2840     SBC #0
6811 85CB       2860     STA SCROLC+1
6813 C943       2880     CMP # >APFDATA-10U ; check underflow
6815 D00E       2900     BNE UPSC10
6817 A5CA       2920     LDA SCROLC
6819 C9AA       2940     CMP # <APFDATA-10U
681B D008       2960     BNE UPSC10
681D A916       2980     LDA # <APFDATA+PFLEN-10U ; reset SCROLC
681F 85CA       3000     STA SCROLC
6821 A946       3020     LDA # >APFDATA+PFLEN-10U
6823 85CB       3040     STA SCROLC+1
                3060 ;
                3080 ; Update DSCROL (new top line)
                3100 ;
6825            3120 UPSC10
6825 CE0840     3140     DEC DSCROL  ; dec index to SMEM first line
6828 1005       3160     BPL UPSC15  ; branch if no underflow
682A A96F       3180     LDA #111    ; else reset to bottom
682C 8D0840     3200     STA DSCROL
                3220 ;
                3240 ; Set pointer to new line destination
                3260 ; which is the new top line.
                3280 ;
682F            3300 UPSC15
```

```
                    3280 ;
682F                3300 UPSC15
```



```
682F AD0840         3320     LDA DSCROL  ; DSCROL indexes top line of display
6832 AA             3340     TAX         ; use to index DSPTB list of ptrs to SMEM
6833 BD3C40         3360     LDA DSPTBL,X ; get pointer low
6836 85D4           3380     STA R+0
6838 BD1B41         3400     LDA DSPTBH,X ;    "    high
683B 85D5           3420     STA R+1
                    3440 ;
                    3460 ; Write PF line indexed by SCROL pointers
                    3480 ; to SMEM address in (R+0,R+1).
                    3500 ;
683D                3520 UPSC18
683D 20C868         3540     JSR PFLINE
                    3560 ;
6840 20E352         3580     JSR CHGDSP  ; rewrite display list pointers
6843 60             3600     RTS
                    3620 ;
                    3640 ;
                    3660 ; SCROLL THE DISPLAY WINDOW DOWN
                    3680 ;
                    3700 ; write new line to current line
                    3720 ;
6844                3740 DNSCRN
6844 AE0840         3760     LDX DSCROL  ; get top line index
6847 BD3C40         3780     LDA DSPTBL,X ; set pointer
684A 85D4           3800     STA R+0
684C BD1B41         3820     LDA DSPTBH,X
684F 85D5           3840     STA R+1
                    3860 ;
                    3880 ;
                    3900 ; set SCROL ptrs to 112th line
                    3920 ; after current line
                    3940 ;
6851 A5CA           3960     LDA SCROLC  ; save SCROL ptrs
6853 48             3980     PHA
6854 A5CB           4000     LDA SCROLC+1
6856 48             4020     PHA
                    4040 ;
6857 A5CA           4060     LDA SCROLC  ; add 70 bytes = 7 char rows = 112 lines
6859 18             4080     CLC
685A 6946           4100     ADC #70
685C 85CA           4120     STA SCROLC
685E A5CB           4140     LDA SCROLC+1
6860 6900           4160     ADC #0
6862 85CB           4180     STA SCROLC+1
                    4200 ;
6864 C946           4220     CMP # >APFDATA+PFLENU ; check ovfl
6866 9015           4240     BCC DNSC02  ; no ovfl - branch
6868 D006           4260     BNE DNSC01  ; ovfl - branch
686A A5CA           4280     LDA SCROLC  ; need to check low byte
686C C920           4300     CMP # <APFDATA+PFLENU
686E 900D           4320     BCC DNSC02  ; no ovfl
                    4340 ;
6870                4360 DNSC01
6870 A5CA           4380     LDA SCROLC
6872 38             4400     SEC
6873 E96C           4420     SBC # <PFLEN
6875 85CA           4440     STA SCROLC
6877 A5CB           4460     LDA SCROLC+1
```

Moore Business Forms, Inc. sv 1412Rh

```
6877 A5CB       4460     LDA SCROLC+1
```



```
6879 E902       4480     SBC # >PFLEN
687B 85CB       4500     STA SCROLC+1
687D            4520 DNSC02
687D 20C868     4540     JSR PFLINE  ; write the line
                4560 ;
6880 68         4580     PLA         ; restore SCROL
6881 85CB       4600     STA SCROLC+1
6883 68         4620     PLA
6884 85CA       4640     STA SCROLC
                4660 ;
                4680 ; Set SCROL ptrs to next line
                4700 ;
6886 AE0B40     4720     LDX SCROLF  ; fine scroll
6889 E8         4740     INX
688A 8E0B40     4760     STX SCROLF
688D E010       4780     CPX #16
688F D024       4800     BNE DNSC10
6891 A900       4820     LDA #0
6893 8D0B40     4840     STA SCROLF  ; reset
6896 A5CA       4860     LDA SCROLC  ; inc coarse
6898 18         4880     CLC
6899 690A       4900     ADC #10
689B 85CA       4920     STA SCROLC
689D A5CB       4940     LDA SCROLC+1
689F 6900       4960     ADC #0
68A1 85CB       4980     STA SCROLC+1
68A3 C946       5000     CMP # >APFDATA+PFLEN; check overflow
68A5 D00E       5020     BNE DNSC10
68A7 A5CA       5040     LDA SCROLC
68A9 C920       5060     CMP # <APFDATA+PFLEN
68AB D008       5080     BNE DNSC10
68AD A9B4       5100     LDA # <PFDATA ; reset
68AF 85CA       5120     STA SCROLC
68B1 A943       5140     LDA # >PFDATA
68B3 85CB       5160     STA SCROLC+1
68B5            5180 DNSC10
                5200 ;
                5220 ; Update DSCROL (new top line)
                5240 ; the old top line now becomes
                5260 ; the new bottom line.
                5280 ;
68B5 EE0840     5300     INC DSCROL  ; inc index to SMEM first line
68B8 AD0840     5320     LDA DSCROL  ; check for overflow
68BB C970       5340     CMP #112
68BD D005       5360     BNE DNSC15
68BF A900       5380     LDA #0      ; reset to top
68C1 8D0840     5400     STA DSCROL
68C4            5420 DNSC15
                5440 ;
68C4 20E352     5460     JSR CHGDSP  ; move the display list pointers
68C7 60         5480     RTS
                5500 ;
                5520 ;
                5540 ;
                5560 ;
                5580 ; Write the playfield line indicated
                5600 ; by SCROLC,SCROLF to the SMEM
                5620 ; location (R+0,R+1).
```

Moore Business Forms, Inc. sv 14128H

```
                5600 ; by SCROLC,SCROLF to the SMEM
                5620 ; location (R+0,R+1).
```



```
                5640 ;
68C8            5660 PFLINE
68C8 8A         5680     TXA          ; save regs
68C9 48         5700     PHA
68CA 98         5720     TYA
68CB 48         5740     PHA
68CC A5D4       5760     LDA R+0
68CE 48         5780     PHA
68CF A5D5       5800     LDA R+1
68D1 48         5820     PHA
68D2 A000       5840     LDY #0       ; index to chars in PF line
68D4            5860 PFLI10
68D4 98         5880     TYA
68D5 48         5900     PHA          ; save char index
68D6 B1CA       5920     LDA (SCROLC),Y
68D8 AA         5940     TAX
68D9 BD0A42     5960     LDA CHARH,X
68DC 85D7       5980     STA R+3
68DE BD4A42     6000     LDA CHARL,X
68E1 AE0B40     6020     LDX SCROLF
68E4 18         6040     CLC
68E5 7DFA41     6060     ADC M4TBL,X
68E8 85D6       6080     STA R+2
68EA A003       6100     LDY #3       ; write 4 bytes/char row
68EC            6120 PFLI20
68EC B1D6       6140     LDA (R+2),Y ; get char data
68EE 91D4       6160     STA (R+0),Y ; write to screen
68F0 88         6180     DEY
68F1 10F9       6200     BPL PFLI20
68F3 A5D4       6220     LDA R+0      ; update screen pointer
68F5 18         6240     CLC
68F6 6904       6260     ADC #4
68F8 85D4       6280     STA R+0
68FA A5D5       6300     LDA R+1
68FC 6900       6320     ADC #0
68FE 85D5       6340     STA R+1
6900 68         6360     PLA          ; restore char counter
6901 A8         6380     TAY
6902 C8         6400     INY
6903 C00A       6420     CPY #10
6905 D0CD       6440     BNE PFLI10  ; do next char
6907 68         6460     PLA
6908 85D5       6480     STA R+1
690A 68         6500     PLA
690B 85D4       6520     STA R+0
690D 68         6540     PLA
690E A8         6560     TAY
690F 68         6580     PLA
6910 AA         6600     TAX
6911 60         6620     RTS
                014560   .INCLUDE #D2:DATEXP.M65
                1000     .PAGE "DATEXP"
```

```
                        1020 ; DATEXP.M65
                        1040 ; DATA EXPANSION ROUTINE
                        1060 ;
                        1080 ;
                        1100 ; ROUTINE EXPANDS TRUCK DATA
                        1120 ; INTO RAM.  TRUCK TYPES TO EXPAND
                        1140 ; ARE PASSED IN TYPREF,TYPREF+1,...
                        1160 ; (# TYPES ALLOWED IS "NACTYPE")
                        1180 ;
                        1200 ; OBJECT DATA AND MASKS ARE THEN
                        1220 ; ACCESSED THROUGH OBJDAT AND OBJMSK
                        1240 ; WHICH ARE STRUCTURED AS FOLLOWS:
                        1260 ;
                        1280 ; TYPE=n1 DIR=0   shift0, shift1, shift2, shift3
                        1300 ;         DIR=1   shift0, shift1, shift2, shift3
                        1320 ;         DIR=3   shift0, shift1, shift2, shift3
                        1340 ;         DIR=4   shift0, shift1, shift2, shift3
                        1360 ; TYPE=n2 DIR=0   shift0, .....
                        1380 ;         DIR=1   .....
                        1382 ;
                        1390 ; Note: Dir 2,5 not used for trucks
                        1400 ;
   6912                 1420 DATEXP
   6912 A900            1440     LDA #0      ; zero OBJREF
   6914 A200            1460     LDX #OBJMAX-1
   6916                 1480 DATE05
   6916 9DC769          1500     STA OBJREF,X
   6919 CA              1520     DEX
   691A 10FA            1540     BPL DATE05
                        1560 ;
   691C A200            1580     LDX #0      ; init index to TYPREF
   691E 86DC            1600     STX R+8
   6920                 1620 DATE10
   6920 BCC569          1640     LDY TYPREF,X ; set ptr to picture pair
   6923 B9C969          1660     LDA PPL,Y
   6926 85D4            1680     STA R+0
   6928 B9D569          1700     LDA PPH,Y
   692B 85D5            1720     STA R+1
                        1740 ;
   692D A900            1760     LDA #0
   692F 85E3            1780     STA R+15    ; set fill value for shift routine
                        1800 ;
   6931                 1820 DATE15
   6931 BDE169          1840     LDA DXPTR1L,X ; set R+2,3 to data area for shifts
   6934 85D6            1860     STA R+2
   6936 BDE569          1880     LDA DXPTR1H,X
   6939 85D7            1900     STA R+3
                        1920 ;
   693B BDE969          1940     LDA DXPTR2L,X ; set R+6,7 to data area for reflection
   693E 85DA            1960     STA R+6
   6940 BDED69          1980     LDA DXPTR2H,X
   6943 85DB            2000     STA R+7
                        2020 ;
   6945 8A              2040     TXA         ; save Xreg.
   6946 48              2060     PHA
   6947 0A              2080     ASL A       ; set ptr to master obj in OBJDAT/OBJMSK
   6948 0A              2100     ASL A
   6949 0A              2120     ASL A
```

```
6948 0A          2100      ASL A
6949 0A          2120      ASL A
```



```
694A 0A          2140      ASL A
694B 0A          2160      ASL A
694C AA          2180      TAX
694D A5D4        2200      LDA R+0
694F 9DF169      2220      STA OBJDAT,X
6952 A5D5        2240      LDA R+1
6954 9DF269      2260      STA OBJDAT+1,X
                 2280 ;
6957 20716A      2300      JSR SHAPEX  ; expand 1st shape
695A A5D4        2320      LDA R+0     ; set ptr to 2nd shape
695C 18          2340      CLC
695D 6950        2360      ADC #80
695F 85D4        2380      STA R+0
6961 9002        2400      BCC DATE20
6963 E6D5        2420      INC R+1
6965             2440 DATE20
6965 A5D6        2441      LDA R+2
6967 85DA        2442      STA R+6
6969 A5D7        2443      LDA R+3
696B 85DB        2444      STA R+7
                 2445 ;
696D A5D6        2446      LDA R+2
696F 38          2447      SEC
6970 E930        2448      SBC # <560
6972 85D6        2449      STA R+2
6974 A5D7        2450      LDA R+3
6976 E902        2451      SBC # >560
6978 85D7        2452      STA R+3
                 2453 ;
697A 20716A      2460      JSR SHAPEX  ; expand 2nd shape
697D 68          2480      PLA         ; recover Xreg
697E 48          2500      PHA
697F 0A          2520      ASL A
6980 0A          2540      ASL A
6981 0A          2560      ASL A
6982 0A          2580      ASL A
6983 0A          2600      ASL A
6984 AA          2620      TAX
6985 A5D4        2640      LDA R+0
6987 9DF969      2660      STA OBJDAT+8,X
698A A5D5        2680      LDA R+1
698C 9DFA69      2700      STA OBJDAT+9,X
698F 68          2720      PLA
6990 AA          2740      TAX
                 2760 ;
6991 E002        2780      CPX #2      ; already do masks?
6993 B016        2800      BCS DATE30  ; yes - branch
6995 BCC569      2820      LDY TYPREF,X ; else setup ptr to mask data
6998 B9CF69      2840      LDA PPL+6,Y
699B 85D4        2860      STA R+0
699D B9DB69      2880      LDA PPH+6,Y
69A0 85D5        2900      STA R+1
69A2 E8          2920      INX
69A3 E8          2940      INX
69A4 A9FF        2960      LDA #$FF    ; fill value for masks
69A6 85E3        2980      STA R+15
69A8 4C3169      3000      JMP DATE15  ; do masks
69AB             3020 DATE30
```

```
69AB A6DC       3030      LDX R+8
69AD BEC569     3040      LDY TYPREF,X ; set OBJREF entry
69B0 8A         3060      TXA
69B1 0A         3080      ASL A        ; *32
69B2 0A         3100      ASL A
69B3 0A         3120      ASL A
69B4 0A         3140      ASL A
69B5 0A         3160      ASL A
69B6 9DC769     3180      STA OBJREF,X
                3200 ;
                3220 ;
69B9 E6DC       3240      INC R+8
69BB A6DC       3260      LDX R+8
69BD E002       3280      CPX #NACTYPE ; # of active object types
69BF F003       3300      BEQ DATE40
69C1 4C2069     3320      JMP DATE10
69C4            3340 DATE40
69C4 60         3360      RTS
                3380 ;
69C5            3400 TYPREF *= *+2
    =0002       3420 NACTYPE = 2      ; # active objects
69C7            3440 OBJREF *= *+OBJMAX
    =2E80       3460 MASKX = $2E80    ; size = 2240 bytes = $8c0
    =3740       3480 DATAX = $3740    ; size= 2240 = $8c0
69C8 FF         3500 FFBYTE .BYTE $FF
                3520 ;
                3540 ;
69C9            3560 PPL
69C9 DF1F1F     3580      .BYTE  <TRUCK, <SWEEPR, <FLGCAR ; data
69CC 1F1F1F     3600      .BYTE  <FLGCAR, <FLGCAR, <FLGCAR
69CF 7F1F1F     3620      .BYTE  <TRUCKM, <SWEEPRM, <FLGCARM ; mask
69D2 1F1F1F     3640      .BYTE  <FLGCARM, <FLGCARM, <FLGCARM
69D5            3660 PPH
69D5 6B6D6D     3680      .BYTE  >TRUCK, >SWEEPR, >FLGCAR ; data
69D8 6D6D6D     3700      .BYTE  >FLGCAR, >FLGCAR, >FLGCAR
69DB 6C6D6D     3720      .BYTE  >TRUCKM, >SWEEPRM, >FLGCARM ; mask
69DE 6D6D6D     3740      .BYTE  >FLGCARM, >FLGCARM, >FLGCARM
69E1            3760 DXPTR1L
69E1 40A0       3780      .BYTE  <DATAX, <DATAX+1120
69E3 80E0       3800      .BYTE  <MASKX, <MASKX+1120
69E5            3820 DXPTR1H
69E5 373B       3840      .BYTE  >DATAX, >DATAX+1120
69E7 2E32       3860      .BYTE  >MASKX, >MASKX+1120
69E9            3880 DXPTR2L
69E9 2080       3900      .BYTE  <DATAX+480, <DATAX+1600
69EB 60C0       3920      .BYTE  <MASKX+480, <MASKX+1600
69ED            3940 DXPTR2H
69ED 393D       3960      .BYTE  >DATAX+480, >DATAX+1600
69EF 3034       3980      .BYTE  >MASKX+480, >MASKX+1600
                4000 ;
                4020 ;
69F1            4040 OBJDAT
69F1 0000       4060      .WORD 0,DATAX,DATAX+80,DATAX+160
69F3 4037
69F5 9037
69F7 E037
69F9 0000       4080      .WORD 0,DATAX+240,DATAX+320,DATAX+400
69FB 3038
```

Moore Business Forms, Inc. s/y 1412RH

```
69F9 0000       4080    .WORD 0,DATAX+240,DATAX+320,DATAX+400
69FB 3038
```



```
69FD 8038
69FF D038
6A01 2039       4100    .WORD DATAX+480,DATAX+560,DATAX+640,DATAX+720
6A03 7039
6A05 C039
6A07 103A
6A09 603A       4120    .WORD DATAX+800,DATAX+880,DATAX+960,DATAX+1040
6A0B B03A
6A0D 003B
6A0F 503B
                4140 ;
6A11 0000       4160    .WORD 0,DATAX+1120,DATAX+1200,DATAX+1280
6A13 A03B
6A15 F03B
6A17 403C
6A19 0000       4180    .WORD 0,DATAX+1360,DATAX+1440,DATAX+1520
6A1B 903C
6A1D E03C
6A1F 303D
6A21 803D       4200    .WORD DATAX+1600,DATAX+1680,DATAX+1760,DATAX+1840
6A23 D03D
6A25 203E
6A27 703E
6A29 C03E       4220    .WORD DATAX+1920,DATAX+2000,DATAX+2080,DATAX+2160
6A2B 103F
6A2D 603F
6A2F B03F
                4240 ;
                4260 ;
6A31            4280 OBJMSK
6A31 0000       4300    .WORD 0,MASKX,MASKX+80,MASKX+160
6A33 802E
6A35 D02E
6A37 202F
6A39 0000       4320    .WORD 0,MASKX+240,MASKX+320,MASKX+400
6A3B 702F
6A3D C02F
6A3F 1030
6A41 6030       4340    .WORD MASKX+480,MASKX+560,MASKX+640,MASKX+720
6A43 B030
6A45 0031
6A47 5031
6A49 A031       4360    .WORD MASKX+800,MASKX+880,MASKX+960,MASKX+1040
6A4B F031
6A4D 4032
6A4F 9032
                4380 ;
6A51 0000       4400    .WORD 0,MASKX+1120,MASKX+1200,MASKX+1280
6A53 E032
6A55 3033
6A57 8033
6A59 0000       4420    .WORD 0,MASKX+1360,MASKX+1440,MASKX+1520
6A5B D033
6A5D 2034
6A5F 7034
6A61 C034       4440    .WORD MASKX+1600,MASKX+1680,MASKX+1760,MASKX+1840
6A63 1035
6A65 6035
```

Moore Business Forms, Inc. sv 1412RH

```
6A63 1035
6A65 6035
```



```
6A67 B035
6A69 0036       4460    .WORD MASKX+1920,MASKX+2000,MASKX+2080,MASKX+2160
6A6B 5036
6A6D A036
6A6F F036
                4480 ;
                4500 ;
                4520 ; EXPAND THE SHAPE POINTED TO BY
                4540 ; R+0,1. R+2,3 POINTS TO SHIFT EXPANSION
                4560 ; AREA, R+6,7 POINTS TO REFLECTION
                4580 ; EXPANSION AREA.
                4600 ;
6A71            4620 SHAPEX
6A71 A5D4       4640    LDA R+0     ; preserve R+0,1
6A73 48         4660    PHA
6A74 A5D5       4680    LDA R+1
6A76 48         4700    PHA
                4720 ;
6A77 A901       4740    LDA #1      ; set flag to do shifts twice
6A79 85ED       4760    STA R+25
6A7B 20FD6A     4780    JSR REFLECT ; reflect the master image
                4800 ;
6A7E            4820 SHAP05
6A7E A903       4840    LDA #3      ; do 3 shifts of master
6A80 85DD       4860    STA R+9
6A82            4880 SHAP10
6A82 20BD6A     4900    JSR SHIFT
                4920 ;
6A85 A5D6       4940    LDA R+2
6A87 85D4       4960    STA R+0
6A89 A5D7       4980    LDA R+3
6A8B 85D5       5000    STA R+1
                5020 ;
6A8D A5D6       5040    LDA R+2
6A8F 18         5060    CLC
6A90 6950       5080    ADC #80
6A92 85D6       5100    STA R+2
6A94 9002       5120    BCC SHAP30
6A96 E6D7       5140    INC R+3
                5160 ;
6A98            5180 SHAP30
6A98 C6DD       5200    DEC R+9
6A9A D0E6       5220    BNE SHAP10
                5240 ;
6A9C C6ED       5260    DEC R+25    ; see if 2nd time through
6A9E 3016       5280    BMI SHAP40  ; yes - exit
6AA0 A5DB       5300    LDA R+7     ; else set ptrs for shifting reflected
                                    image
6AA2 85D5       5320    STA R+1
6AA4 A5DA       5340    LDA R+6
6AA6 85D4       5360    STA R+0
6AA8 18         5380    CLC
6AA9 6950       5400    ADC #80
6AAB 85D6       5420    STA R+2
6AAD A5D5       5440    LDA R+1
6AAF 6900       5460    ADC #0
6AB1 85D7       5480    STA R+3
6AB3 4C7E6A     5500    JMP SHAP05
```

Moore Business Forms, Inc. sv 1412RH

```
6AAF 6900       5460      ADC #0
6AB1 85D7       5480      STA R+3
6AB3 4C7E6A     5500      JMP SHAP05
```



```
                5520 ;
6AB6            5540 SHAP40
6AB6 68         5560      PLA
6AB7 85D5       5580      STA R+1
6AB9 68         5600      PLA
6ABA 85D4       5620      STA R+0
6ABC 60         5640      RTS
                5660 ;
                5680 ;
                5700 ; SHIFT DATA 1 PIXEL TO THE RIGHT
                5720 ; SOURCE IS R+0,1 DESTINATION IS R+2,3
                5740 ; Fill byte is in R+15 (value shifted
                5760 ; in from left).
                5780 ;
6ABD            5800 SHIFT
6ABD A903       5820      LDA #3
6ABF 85D9       5840      STA R+5
6AC1            5860 SHIF10
6AC1 A000       5880      LDY #0
6AC3            5900 SHIF15
6AC3 A200       5920      LDX #0        ; copy row to registers
6AC5            5940 SHIF20
6AC5 B1D4       5960      LDA (R+0),Y
6AC7 95DE       5980      STA R+10,X
6AC9 C8         6000      INY
6ACA E8         6020      INX
6ACB E005       6040      CPX #5
6ACD D0F6       6060      BNE SHIF20
6ACF A5E3       6080      LDA R+15
                6100 ;
6AD1 6A         6120      ROR A
6AD2 66DE       6140      ROR R+10      ; shift row right by 1 pixel
6AD4 66DF       6160      ROR R+11
6AD6 66E0       6180      ROR R+12
6AD8 66E1       6200      ROR R+13
6ADA 66E2       6220      ROR R+14
6ADC 6A         6240      ROR A
6ADD 66DE       6260      ROR R+10
6ADF 66DF       6280      ROR R+11
6AE1 66E0       6300      ROR R+12
6AE3 66E1       6320      ROR R+13
6AE5 66E2       6340      ROR R+14
                6360 ;
6AE7 98         6380      TYA           ; rewrite registers to dest. row
6AE8 38         6400      SEC
6AE9 E905       6420      SBC #5
6AEB A8         6440      TAY
                6460 ;
6AEC A200       6480      LDX #0
6AEE            6500 SHIF30
6AEE B5DE       6520      LDA R+10,X
6AF0 91D6       6540      STA (R+2),Y
6AF2 C8         6560      INY
6AF3 E8         6580      INX
6AF4 E005       6600      CPX #5
6AF6 D0F6       6620      BNE SHIF30
                6640 ;
6AF8 C050       6660      CPY #80       ; do for 80 bytes (16 rows)
```

Moore Business Forms, Inc. sv 1412RM

```
6AFA D0C7       6680      BNE SHIF15
                6700 ;
6AFC 60         6720      RTS
                6740 ;
                6760 ;
                6780 ; REFLECT DATA AT R+0,1 TO R+6,7
                6782 ; LAST COLUMN OF BYTES IS NOT
                6784 ; REFLECTED
                6800 ;
6AFD            6820 REFLECT
6AFD A04F       6840      LDY #79     ; reflect 80 bytes
6AFF            6860 REFL10
6AFF A900       6880      LDA #0
6B01 85D8       6900      STA R+4
                6920 ;
6B03 B1D4       6940      LDA (R+0),Y ; get byte to reflect
                6960 ;
                6980 ; Pixel bits must be switched if
                7000 ; different since ROL will switch
                7020 ; them otherwise
                7040 ;
6B05 A204       7060      LDX #4      ; reflect 4 pixels/byte
6B07            7080 REFL20
6B07 85EC       7100      STA R+24
6B09 24EC       7120      BIT R+24    ; set N,V flags
6B0B 1005       7140      BPL REFL30  ; are pixel bits the same?
6B0D 7007       7160      BVS REFL50
6B0F 4C146B     7180      JMP REFL40
6B12            7200 REFL30
6B12 5002       7220      BVC REFL50
6B14            7240 REFL40
6B14 49C0       7260      EOR #$C0    ; need to reverse pixel
6B16            7280 REFL50
6B16 0A         7300      ASL A
6B17 66D8       7320      ROR R+4
6B19 0A         7340      ASL A
6B1A 66D8       7360      ROR R+4
6B1C CA         7380      DEX
6B1D D0E8       7400      BNE REFL20
                7420 ;
6B1F 98         7440      TYA
6B20 48         7460      PHA
6B21 B92F6B     7480      LDA RFLPAT,Y
6B24 A8         7500      TAY
                7520 ;
6B25 A5D8       7540      LDA R+4
6B27 91DA       7560      STA (R+6),Y
                7580 ;
6B29 68         7600      PLA
6B2A A8         7620      TAY
6B2B 88         7640      DEY
6B2C 10D1       7660      BPL REFL10
                7680 ;
6B2E 60         7700      RTS
                7720 ;
                7740 ; Reflected byte pattern
6B2F            7760 RFLPAT
6B2F 03020100   8081      .BYTE 3,2,1,0,4
```


Moore Business Forms, Inc. sv 1413KH

```
                        7680 ;
6B2E 60                 7700     RTS
                        7720 ;
                        7740 ; Reflected byte pattern
6B2F                    7760 RFLPAT
6B2F 03020100           8081     .BYTE 3,2,1,0,4
```



```
6B33 04
6B34 08070605           8082     .BYTE 8,7,6,5,9
6B38 09
6B39 0D0C0B0A           8083     .BYTE 13,12,11,10,14
6B3D 0E
6B3E 1211100F           8084     .BYTE 18,17,16,15,19
6B42 13
6B43 17161514           8085     .BYTE 23,22,21,20,24
6B47 18
6B48 1C1B1A19           8086     .BYTE 28,27,26,25,29
6B4C 1D
6B4D 21201F1E           8087     .BYTE 33,32,31,30,34
6B51 22
6B52 26252423           8088     .BYTE 38,37,36,35,39
6B56 27
6B57 2B2A2928           8089     .BYTE 43,42,41,40,44
6B5B 2C
6B5C 302F2E2D           8090     .BYTE 48,47,46,45,49
6B60 31
6B61 35343332           8091     .BYTE 53,52,51,50,54
6B65 36
6B66 3A393837           8092     .BYTE 58,57,56,55,59
6B6A 3B
6B6B 3F3E3D3C           8093     .BYTE 63,62,61,60,64
6B6F 40
6B70 44434241           8094     .BYTE 68,67,66,65,69
6B74 45
6B75 49484746           8095     .BYTE 73,72,71,70,74
6B79 4A
6B7A 4E4D4C4B           8096     .BYTE 78,77,76,75,79
6B7E 4F
                        8100 ;
                        8120 ;
                        8140 ; BACKSAVE DATA ADDRESS TABLE
                        8160 ;
6B7F                    8180 BSTBLL ;        low
6B7F 8FDF2F7F           8200     .BYTE  <BAKSAV, <BAKSAV+80, <BAKSAV+160, <BAKSAV+240
6B83 CF1F6FBF           8220     .BYTE  <BAKSAV+320, <BAKSAV+400, <BAKSAV+480,
                                        <BAKSAV+560
6B87                    8240 BSTBLH ;        high
6B87 6B6B6C6C           8260     .BYTE  >BAKSAV, >BAKSAV+80, >BAKSAV+160, >BAKSAV+240
6B8B 6C6D6D6D           8280     .BYTE  >BAKSAV+320, >BAKSAV+400, >BAKSAV+480,
                                        >BAKSAV+560
                        8300 ;
                        8320 ; BACKSAVE DATA AREA
                        8340 ;
6B8F                    8360 BAKSAV *= *+OBJMAX*80
                        014580   .INCLUDE #D2:TRUCKS.M65
                        1000 ; TRUCK DATA  POS=0
6BDF                    1005 TRUCK
6BDF 00000000           1010     .BYTE $00,$00,$00,$00,$00
6BE3 00
6BE4 00000000           1020     .BYTE $00,$00,$00,$00,$00
6BE8 00
6BE9 00055400           1030     .BYTE $00,$05,$54,$00,$00
6BED 00
6BEE 00155500           1040     .BYTE $00,$15,$55,$00,$00
6BF2 00
```

Moore Business-Forms, Inc. sv 1412RH

```
6BEE 00155500   1040      .BYTE $00,$15,$55,$00,$00
6BF2 00
```



```
DATEXP                                                  D2:TRUCK6.M65


6BF3 005FFD00   1050      .BYTE $00,$5F,$FD,$00,$00
6BF7 00
6BF8 15DFFD00   1060      .BYTE $15,$DF,$FD,$00,$00
6BFC 00
6BFD 55D55540   1070      .BYTE $55,$D5,$55,$40,$00
6C01 00
6C02 955FFF54   1080      .BYTE $95,$5F,$FF,$54,$00
6C06 00
6C07 9557FFF5   1090      .BYTE $95,$57,$FF,$F5,$00
6C0B 00
6C0C 9555FFFD   1100      .BYTE $95,$55,$FF,$FD,$00
6C10 00
6C11 1A555555   1110      .BYTE $1A,$55,$55,$55,$00
6C15 00
6C16 2A955555   1120      .BYTE $2A,$95,$55,$55,$00
6C1A 00
6C1B 00AAAAAA   1130      .BYTE $00,$AA,$AA,$AA,$00
6C1F 00
6C20 00A56A95   1140      .BYTE $00,$A5,$6A,$95,$00
6C24 00
6C25 00AA95AA   1150      .BYTE $00,$AA,$95,$AA,$00
6C29 00
6C2A 002A002A   1160      .BYTE $00,$2A,$00,$2A,$00
6C2E 00
                1170 ;  POS=1
6C2F 00015500   1180      .BYTE $00,$01,$55,$00,$00
6C33 00
6C34 00055540   1190      .BYTE $00,$05,$55,$40,$00
6C38 00
6C39 0017FF40   1200      .BYTE $00,$17,$FF,$40,$00
6C3D 00
6C3E 0077FF40   1210      .BYTE $00,$77,$FF,$40,$00
6C42 00
6C43 05755540   1220      .BYTE $05,$75,$55,$40,$00
6C47 00
6C48 155FFF50   1230      .BYTE $15,$5F,$FF,$50,$00
6C4C 00
6C4D 255FFF54   1240      .BYTE $25,$5F,$FF,$54,$00
6C51 00
6C52 2557FFD4   1250      .BYTE $25,$57,$FF,$D4,$00
6C56 00
6C57 2957FFD5   1260      .BYTE $29,$57,$FF,$D5,$00
6C5B 00
6C5C 2955FFF5   1270      .BYTE $29,$55,$FF,$F5,$00
6C60 00
6C61 0055FFF5   1280      .BYTE $00,$55,$FF,$F5,$00
6C65 00
6C66 00555555   1290      .BYTE $00,$55,$55,$55,$00
6C6A 00
6C6B 00555555   1300      .BYTE $00,$55,$55,$55,$00
6C6F 00
6C70 00AAAAAA   1310      .BYTE $00,$AA,$AA,$AA,$00
6C74 00
6C75 00AA80AA   1320      .BYTE $00,$AA,$80,$AA,$00
6C79 00
6C7A 002A002A   1330      .BYTE $00,$2A,$00,$2A,$00
6C7E 00
                1350 ; TRUCK DATA  POS=0
```

Moore Business Forms, Inc. #v 14128H

```
6C7E 00
                1350 ; TRUCK DATA  POS=0


PAGE 74   UPN DOWN - for Atari systems
DATEXP                                                  D2:TRUCKS.M65

6C7F            1355 TRUCKM
6C7F FFFFFFFF   1360     .BYTE $FF,$FF,$FF,$FF,$FF
6C83 FF
6C84 FFFFFFFF   1370     .BYTE $FF,$FF,$FF,$FF,$FF
6C88 FF
6C89 FFF003FF   1380     .BYTE $FF,$F0,$03,$FF,$FF
6C8D FF
6C8E FFC000FF   1390     .BYTE $FF,$C0,$00,$FF,$FF
6C92 FF
6C93 FF0000FF   1400     .BYTE $FF,$00,$00,$FF,$FF
6C97 FF
6C98 C00000FF   1410     .BYTE $C0,$00,$00,$FF,$FF
6C9C FF
6C9D 0000003F   1420     .BYTE $00,$00,$00,$3F,$FF
6CA1 FF
6CA2 00000003   1430     .BYTE $00,$00,$00,$03,$FF
6CA6 FF
6CA7 00000000   1440     .BYTE $00,$00,$00,$00,$FF
6CAB FF
6CAC 00000000   1450     .BYTE $00,$00,$00,$00,$FF
6CB0 FF
6CB1 C0000000   1460     .BYTE $C0,$00,$00,$00,$FF
6CB5 FF
6CB6 C0000000   1470     .BYTE $C0,$00,$00,$00,$FF
6CBA FF
6CBB FF000000   1480     .BYTE $FF,$00,$00,$00,$FF
6CBF FF
6CC0 FF000000   1490     .BYTE $FF,$00,$00,$00,$FF
6CC4 FF
6CC5 FF000000   1500     .BYTE $FF,$00,$00,$00,$FF
6CC9 FF
6CCA FFC0FFC0   1510     .BYTE $FF,$C0,$FF,$C0,$FF
6CCE FF
                1520 ; POS=1
6CCF FFFC00FF   1530     .BYTE $FF,$FC,$00,$FF,$FF
6CD3 FF
6CD4 FFF0003F   1540     .BYTE $FF,$F0,$00,$3F,$FF
6CD8 FF
6CD9 FFC0003F   1550     .BYTE $FF,$C0,$00,$3F,$FF
6CDD FF
6CDE FF00003F   1560     .BYTE $FF,$00,$00,$3F,$FF
6CE2 FF
6CE3 F000003F   1570     .BYTE $F0,$00,$00,$3F,$FF
6CE7 FF
6CE8 C000000F   1580     .BYTE $C0,$00,$00,$0F,$FF
6CEC FF
6CED C0000003   1590     .BYTE $C0,$00,$00,$03,$FF
6CF1 FF
6CF2 C0000003   1600     .BYTE $C0,$00,$00,$03,$FF
6CF6 FF
6CF7 C0000000   1610     .BYTE $C0,$00,$00,$00,$FF
6CFB FF
6CFC C0000000   1620     .BYTE $C0,$00,$00,$00,$FF
6D00 FF
6D01 FF000000   1630     .BYTE $FF,$00,$00,$00,$FF
6D05 FF
6D06 FF000000   1640     .BYTE $FF,$00,$00,$00,$FF
6D0A FF
```

Moore Business Forms, Inc. SV 14128H

```
6D0B FF000000   1650      .BYTE $FF,$00,$00,$00,$FF
6D0F FF
6D10 FF000000   1660      .BYTE $FF,$00,$00,$00,$FF
6D14 FF
6D15 FF003F00   1670      .BYTE $FF,$00,$3F,$00,$FF
6D19 FF
6D1A FFC0FFC0   1680      .BYTE $FF,$C0,$FF,$C0,$FF
6D1E FF
                1690 ;  END OF TRUCK DATA
                1700 ;
6D1F            1710 SWEEPR
6D1F            1720 SWEEPRM
6D1F            1730 FLGCAR
6D1F            1740 FLGCARM
                1750 ;
                1760 ;
                014600   .INCLUDE #D2:FLAGS.M65
                1000     .PAGE "FLAGS.M65"
```

Moore Business Forms, Inc. sv 1412RN

```
FLAGS.M65                                D2:FLAGS.M65


                    1020 ;
                    1040 ; FLAGS.M65
                    1060 ; FLAG ROUTINES
                    1080 ;
6D1F                1100 FTOGL *= *+1     ; enable flag toggles
6D20                1120 FHIT *= *+1      ; last flag hit or currently hit
6D21                1140 FDLYCT *= *+1    ; flag delay ctr
     =0050          1160 FDLY =  80       ; delay betw flag hit checks
     =D00B          1180 M3PL =  $D00B    ; missile to player hit
     =D01E          1200 HITCLR = $D01E   ; clear hits write location
                    1220 ;
                    1240 ; *FLGHIT*
                    1260 ; HANDLES FLAG TOGGLING
                    1280 ;
6D22                1300 FLGHIT
6D22 CE216D         1320      DEC FDLYCT
6D25 F003           1340      BEQ FLGH04
6D27 4CCD6D         1360      JMP FLGH90
6D2A                1380 FLGH04
6D2A A950           1400      LDA #FDLY   ; reset delay
6D2C 8D216D         1420      STA FDLYCT
6D2F AD0BD0         1440      LDA M3PL    ; any hits?
6D32 D003           1460      BNE FLGH06
6D34 4CC86D         1480      JMP FLGH80  ; no - branch
6D37                1500 FLGH06
6D37 8D1ED0         1520      STA HITCLR  ; clear hits
                    1540 ;
                    1560 ; Find out which flag is hit
                    1580 ;
6D3A A209           1600      LDX #9
6D3C                1620 FLGH10
6D3C BDD86D         1640      LDA FYH,X   ; can't have MSB of Y <> 0
6D3F D028           1660      BNE FLGH40
6D41 BDCE6D         1680      LDA FYL,X   ; check Y
6D44 38             1700      SEC
6D45 E904           1720      SBC #4
6D47 EDA742         1740      SBC BUGY
6D4A 1005           1760      BPL FLGH20
6D4C 49FF           1780      EOR ##$FF
6D4E 18             1800      CLC
6D4F 6901           1820      ADC #1
6D51                1840 FLGH20
6D51 C909           1860      CMP #9
6D53 B014           1880      BCS FLGH40
6D55 BDEC6D         1900      LDA FHPOS,X ; check x
6D58 38             1920      SEC         ; convert from P/M
6D59 E934           1940      SBC #52     ; won't ovfl
6D5B EDA642         1960      SBC BUGX
6D5E 1005           1980      BPL FLGH30
6D60 49FF           2000      EOR ##$FF
6D62 18             2020      CLC
6D63 6901           2040      ADC #1
6D65                2060 FLGH30
6D65 C909           2080      CMP #9
6D67 9006           2100      BCC FLGH50
6D69                2120 FLGH40
6D69 CA             2140      DEX         ; next flag
6D6A 10D0           2160      BPL FLGH10
```

Moore Business Forms, Inc. xv 1412RH

```
6D69                2120 FLGH40
6D69 CA             2140    DEX         ; next flag
6D6A 10D0           2160    BPL FLGH10
```



```
6D6C 4CCD6D         2180    JMP FLGH90  ; error exit
                    2200 ;
                    2220 ; Got flag
                    2240 ;
6D6F                2260 FLGH50
6D6F EC206D         2280    CPX FHIT    ; is this hit a continuation of last?
6D72 F059           2300    BEQ FLGH90  ; yes - branch
6D74 8E206D         2320    STX FHIT    ; save currnet flag hit
6D77 BDE26D         2340    LDA FCOLOR,X ; is flag on or off?
6D7A 2901           2360    AND #1
6D7C F005           2380    BEQ FLGH60  ; on - branch
6D7E AD1F6D         2400    LDA FTOGL   ; toggle enabled?
6D81 F04A           2420    BEQ FLGH90  ; no - branch
6D83                2440 FLGH60
6D83 200B66         2460    JSR FLAG    ; do flag sound
6D86 BDE26D         2480    LDA FCOLOR,X ; change flag
6D89 4901           2500    EOR #1
6D8B 9DE26D         2520    STA FCOLOR,X
                    2540 ;
                    2560 ;
                    2580 ; Change header display
                    2600 ;
6D8E E005           2620    CPX #5      ; is header flag P/M or PF?
6D90 9021           2640    BCC FLGH70  ; P/M - branch
                    2660 ;
                    2680 ; Flag is Playfield flag
                    2700 ;
6D92 BDF66D         2720    LDA IHFPFL,X ; set dest ptr
6D95 85D4           2740    STA R+0
6D97 BD006E         2760    LDA IHFPFH,X
6D9A 85D5           2780    STA R+1
6D9C BDE26D         2800    LDA FCOLOR,X ; which color?
6D9F 2901           2820    AND #1
6DA1 F005           2840    BEQ FLGH62
6DA3 A955           2860    LDA #$55    ; make white
6DA5 4CAB6D         2880    JMP FLGH64
6DA8                2900 FLGH62
6DA8 BD0A6E         2920    LDA IHFPFC,X ; make color
6DAB                2940 FLGH64
6DAB 85D7           2960    STA R+3
6DAD 20FB54         2980    JSR EFLAG   ; draw flag
6DB0 4CCD6D         3000    JMP FLGH90
                    3020 ;
                    3040 ; Flag is a P/M flag
                    3060 ;
6DB3                3080 FLGH70
6DB3 BDE26D         3100    LDA FCOLOR,X
6DB6 2901           3120    AND #1
6DB8 F005           3140    BEQ FLGH74
6DBA A900           3160    LDA #0      ; move flag off screen to reveal white
6DBC 4CC26D         3180    JMP FLGH76
6DBF                3200 FLGH74
6DBF BD4E6E         3220    LDA IHFPM,X ; move back on screen
6DC2                3240 FLGH76
6DC2 9D536E         3260    STA HFPOS,X
6DC5 4CCD6D         3280    JMP FLGH90
                    3300 ;
6DC8                3320 FLGH80
```

```
                        3300 ;
6DC8                    3320 FLGH80






6DC8 A9FF               3340     LDA #$FF
6DCA 8D206D             3360     STA FHIT
6DCD                    3380 FLGH90
6DCD 60                 3400     RTS
                        3420 ;
                        3440 ;
6DCE                    3460 FYL ;          flag Ypos high byte
6DCE C797E787           3480     .BYTE  <455, <407, <743, <647, <615, <847, <791, <327,
                                        <151, <119
6DD2 674F1747
6DD6 9777
6DD8                    3500 FYH ;          "      low   "
6DD8 01010202           3520     .BYTE  >455, >407, >743, >647, >615, >847, >791, >327,
                                        >151, >119
6DDC 02030301
6DE0 0000
6DE2                    3540 FCOLOR ;       flag color byte
6DE2 164A3656           3560     .BYTE $16,$4A,$36,$56,$AA
6DE6 AA
6DE7 861E861E           3580     .BYTE $86,$1E,$86,$1E,$86
6DEB 86
6DEC                    3600 FHPOS ;        (#PFLINES=983)flag horiz. pos.
6DEC 689888A8           3620     .BYTE 104,152,136,168,98
6DF0 62
6DF1 A6588546           3640     .BYTE 166,88,181,70,168
6DF5 A8
                        3660 ;
6DF6                    3680 IHFPFL ;       PF flags horiz pos, low
6DF6 0E12161A           3700     .BYTE  <HMEM+90, <HMEM+90+4, <HMEM+90+8, <HMEM+90+12,
                                        <HMEM+90+16
6DFA 1E
6DFB 1014181C           3720     .BYTE  <HMEM+90+2, <HMEM+90+6, <HMEM+90+10,
                                        <HMEM+90+14, <HMEM+90+18
6DFF 20
6E00                    3740 IHFPFH ;       high
6E00 48484848           3760     .BYTE  >HMEM+90, >HMEM+90+4, >HMEM+90+8, >HMEM+90+12,
                                        >HMEM+90+16
6E04 48
6E05 48484848           3780     .BYTE  >HMEM+90+2, >HMEM+90+6, >HMEM+90+10,
                                        >HMEM+90+14, >HMEM+90+18
6E09 48
6E0A                    3800 IHFPFC ;       playfield flag color
6E0A 55555555           3820     .BYTE $55,$55,$55,$55,$55
6E0E 55
6E0F AAFFAAFF           3840     .BYTE $AA,$FF,$AA,$FF,$AA
6E13 AA
                        3860 ;
                        3880 ;
                        3900 ; SET UP THE FLAGS IN THE HEADER
                        3920 ;
6E14                    3940 IHFLGS
6E14 A20F               3960     LDX #15     ; init P/M data
6E16                    3980 IHFL10
6E16 BDA96F             4000     LDA MFLGDAT+2,X
6E19 9D3273             4020     STA M0+50,X
6E1C 9D3274             4040     STA P0+50,X
6E1F 9D3275             4060     STA P1+50,X
6E22 9D3276             4080     STA P2+50,X
```

Moore Business Forms, Inc. sv 1412RM

```
6E22 9D3276     4080     STA P2+50,X

6E25 9D3277     4100     STA P3+50,X
6E28 CA         4120     DEX
6E29 10EB       4140     BPL IHFL10
                4160 ;
6E2B A204       4180     LDX #4      ; init HPOS data
6E2D            4200 IHFL20
6E2D BD4E6E     4220     LDA IHFPM,X
6E30 9D536E     4240     STA HFPOS,X
6E33 CA         4260     DEX
6E34 10F7       4280     BPL IHFL20
                4300 ;
6E36 A209       4320     LDX #9      ; init 10 header flags
6E38            4340 IHFL30
6E38 BDF66D     4360     LDA IHFPFL,X
6E3B 85D4       4380     STA R+0
6E3D BD006E     4400     LDA IHFPFH,X
6E40 85D5       4420     STA R+1
6E42 BD0A6E     4440     LDA IHFPFC,X
6E45 85D7       4460     STA R+3
6E47 20FB54     4480     JSR EFLAG   ; draw flag
6E4A CA         4500     DEX
6E4B 10EB       4520     BPL IHFL30
                4540 ;
6E4D 60         4560     RTS
                4580 ;
                4600 ; Initial header flag horiz pos.
                4620 ; For P/M flags
6E4E            4640 IHFPM
6E4E 58687888   4660     .BYTE 88,104,120,136,152
6E52 98
6E53            4680 HFPOS *= *+5    ; horiz pos of header P/M flags
                4700 ;
                4720 ;
                4740 ; Scroll the flags in the flag tables
                4760 ;
6E58            4780 FLGSCR
6E58 AD135B     4800     LDA SCRFLG  ; only do this after screen scroll
6E5B F043       4820     BEQ FLGS45
                4840 ;
6E5D A96E       4860     LDA #110
6E5F            4880 FLGS01
6E5F CD0BD4     4900     CMP VCOUNT  ; sync to screen
6E62 B0FB       4920     BCS FLGS01
                4940 ;
                4960 ;
                4980 ; Setup for RSTDLI routine in
                5000 ; case it's called
                5020 ;
6E64 A900       5040     LDA #0      ; unflag 1st DLI
6E66 8DFB6F     5060     STA FTOPDLI
                5080 ;
6E69 A9FF       5100     LDA #$FF    ; init list w/large #
6E6B A204       5120     LDX #MAXFON
6E6D            5140 RSTD08
6E6D 9DF56F     5160     STA FDLISEQ,X
6E70 CA         5180     DEX
6E71 10FA       5200     BPL RSTD08
                5220 ;
```


Moore Business Forms, Inc sv 1412RH

```
6E6D                5140 RSTD08
6E6D 9DF56F         5160     STA FDLISEQ,X
6E70 CA             5180     DEX
6E71 10FA           5200     BPL RSTD08
                    5220 ;
```



```
6E73 AD135B         5240     LDA SCRFLG  ; which way was scroll?
6E76 3029           5260     BMI FLGS50
                    5280 ;
                    5300 ; Scroll flag data down
                    5320 ;
6E78 A209           5340     LDX #9      ; index to flags
6E7A                5360 FLGS10
6E7A FECE6D         5380     INC FYL,X   ; chg flag's Y
6E7D D003           5400     BNE FLGS15
6E7F FED86D         5420     INC FYH,X
6E82                5440 FLGS15
6E82 BDD86D         5460     LDA FYH,X   ; check for wrap
6E85 C903           5480     CMP # >APFLEN/100*16
6E87 9011           5500     BCC FLGS30
6E89 D007           5520     BNE FLGS20
6E8B BDCE6D         5540     LDA FYL,X
6E8E C9E0           5560     CMP # <APFLEN/100*16
6E90 9008           5580     BCC FLGS30
6E92                5600 FLGS20
6E92 A900           5620     LDA #0      ; do wraparound
6E94 9DCE6D         5640     STA FYL,X
6E97 9DD86D         5660     STA FYH,X
6E9A                5680 FLGS30
6E9A 20C86E         5700     JSR DRWFLG
6E9D                5720 FLGS40
6E9D CA             5740     DEX
6E9E 10DA           5760     BPL FLGS10
6EA0                5820 FLGS45
6EA0 60             5840     RTS
                    5860 ;
                    5880 ; Scroll flag data up
                    5900 ;
6EA1                5920 FLGS50
6EA1 A209           5940     LDX #9
6EA3                5960 FLGS55
6EA3 DECE6D         5980     DEC FYL,X
6EA6 BDCE6D         6000     LDA FYL,X
6EA9 C9FF           6020     CMP #$FF
6EAB D003           6040     BNE FLGS60
6EAD DED86D         6060     DEC FYH,X
6EB0                6080 FLGS60
6EB0 BDD86D         6100     LDA FYH,X   ; check for wrap
6EB3 C9FF           6120     CMP #$FF
6EB5 D00A           6140     BNE FLGS70
6EB7 A9DF           6160     LDA # <APFLEN/100*16-1 ; do wraparound
6EB9 9DCE6D         6180     STA FYL,X
6EBC A903           6200     LDA # >APFLEN/100*16-1
6EBE 9DD86D         6220     STA FYH,X
6EC1                6240 FLGS70
6EC1 20C86E         6260     JSR DRWFLG
6EC4                6280 FLGS80
6EC4 CA             6300     DEX
6EC5 10DC           6320     BPL FLGS55
6EC7 60             6380     RTS
                    6400 ;
                    6420 ;
                    6440 ; Draw the flag as a 5th player
                    6460 ; by combining all missiles
```

Moore Business Forms, Inc. 8V 1412RH

```
                6440 ; Draw the flag as a 5th player
                6460 ; by combining all missiles
```



```
                6480 ; uses R+10,11,12,13.  Set DLI
                6500 ;
6EC8            6520 DRWFLG
6EC8 8A         6540     TXA
6EC9 48         6560     PHA
6ECA BDD86D     6580     LDA FYH,X   ; check within display bounds
6ECD D070       6600     BNE DRWF50
6ECF BDCE6D     6620     LDA FYL,X
6ED2 C90B       6640     CMP #11
6ED4 9069       6660     BCC DRWF50
6ED6 C963       6680     CMP #99
6ED8 B065       6700     BCS DRWF50
                6720 ;
6EDA 85DF       6740     STA R+11    ; save flag's Y
6EDC 86E0       6760     STX R+12    ; save flag #
6EDE C913       6780     CMP #19     ; clip top?
6EE0 9036       6800     BCC DRWF30  ; yes - branch
6EE2 C95C       6820     CMP #92     ; clip bottom?
6EE4 B048       6840     BCS DRWF40  ; yes - branch
                6860 ;
                6880 ;
6EE6 A912       6900     LDA #18     ; # bytes to draw
6EE8 85DE       6920     STA R+10
6EEA BDCE6D     6940     LDA FYL,X   ; set index to dest.
6EED A200       6960     LDX #0      ; index to source data
6EEF            6980 DRWF05
6EEF 0A         7000     ASL A       ; convert to P/M coords
6EF0 18         7020     CLC
6EF1 691D       7040     ADC #29
6EF3 A8         7060     TAY
6EF4            7080 DRWF20
6EF4 98         7100     TYA         ; sync to screen
6EF5 4A         7120     LSR A
6EF6 38         7140     SEC
6EF7 E903       7160     SBC #3
6EF9 CD0BD4     7180     CMP VCOUNT
6EFC B008       7200     BCS DRWF22
6EFE 18         7220     CLC
6EFF 6906       7240     ADC #6
6F01            7246 DRWF21
6F01 CD0BD4     7260     CMP VCOUNT
6F04 B0FB       7262     BCS DRWF21
6F06            7264 DRWF22
6F06 BDA76F     7280     LDA MFLGDAT,X ; draw flag
6F09 990073     7300     STA M0,Y
6F0C E8         7320     INX
6F0D C8         7340     INY
6F0E C6DE       7360     DEC R+10
6F10 D0F4       7380     BNE DRWF22
                7400 ;
6F12 20426F     7420     JSR RSTDLI  ; set DLI
                7440 ;
6F15 4C3F6F     7460     JMP DRWF50
                7480 ;
                7500 ; Clip top
                7520 ;
6F18            7540 DRWF30
6F18 A913       7560     LDA #19
```

Moore Business Forms, Inc. sv 1412RH

```
6F18                7540 DRWF30
6F18 A913           7560     LDA #19
```



```
6F1A 38             7580     SEC
6F1B FDCE6D         7600     SBC FYL,X
6F1E 0A             7620     ASL A
6F1F 85DE           7640     STA R+10
6F21 AA             7660     TAX
6F22 A912           7680     LDA #18
6F24 38             7700     SEC
6F25 E5DE           7720     SBC R+10
6F27 85DE           7740     STA R+10
6F29 A043           7760     LDY #67
6F2B 4CF46E         7780     JMP DRWF20
                    7800 ;
                    7820 ; Clip bottom
                    7840 ;
6F2E                7860 DRWF40
6F2E A963           7880     LDA #99
6F30 38             7900     SEC
6F31 FDCE6D         7920     SBC FYL,X
6F34 0A             7940     ASL A
6F35 85DE           7960     STA R+10
6F37 BDCE6D         7980     LDA FYL,X
6F3A A200           8000     LDX #0
6F3C 4CEF6E         8020     JMP DRWF05
6F3F                8040 DRWF50
6F3F 68             8060     PLA
6F40 AA             8080     TAX
6F41 60             8100     RTS
                    8120 ;
                    8140 ;
                    8160 ; Reset flag DLI's to control
                    8180 ; flag's color and HPOS
                    8200 ; R+11    flag's Y
                    8220 ; R+12    flag #
                    8240 ; R+13    used
6F42                8260 RSTDLI
                    8280 ;
6F42 A5DF           8300     LDA R+11     ; compute index to DLIST
6F44 38             8320     SEC          ; based on flag's Y
6F45 E913           8340     SBC #19
6F47 A8             8360     TAY
6F48 B93A4B         8380     LDA M3TBL,Y ; *3
6F4B A8             8400     TAY
6F4C A5DF           8420     LDA R+11     ; top line or above?
6F4E C914           8440     CMP #20
6F50 9005           8460     BCC RSTD10   ; yes - branch
6F52 A96D           8480     LDA #$6D     ; clear top adjacent DLI
6F54 99BE42         8500     STA DLIST+25-3,Y
6F57                8520 RSTD10
6F57 A5DF           8540     LDA R+11     ; top clip?
6F59 C913           8560     CMP #19
6F5B B00D           8580     BCS RSTD30   ; no - branch
6F5D A901           8600     LDA #1       ; set top DLI flag
6F5F 8DFB6F         8620     STA FTOPDLI
6F62 A96D           8640     LDA #$6D     ; erase last DLI
6F64 8DC142         8660     STA DLIST+25
6F67 4C7A6F         8680     JMP RSTD32
6F6A                8700 RSTD30
6F6A A5DF           8720     LDA R+11     ; bottom line or below?
```

```
                    LDA R+11    ; bottom line or below?
```



```
6F6C C962       8740     CMP #98
6F6E B005       8760     BCS RSTD31  ; yes - branch
6F70 A96D       8780     LDA #$6D    ; clear bottom adjacent DLI
6F72 99C442     8800     STA DLIST+25+3,Y
6F75            8820 RSTD31
6F75 A9ED       8840     LDA #$ED    ; set DLI
6F77 99C142     8860     STA DLIST+25,Y
6F7A            8880 RSTD32
                8900 ;
                8920 ; Put flag in FDLISEQ list in order
                8940 ; of display
                8960 ;
6F7A A000       8980     LDY #0
6F7C            9000 RSTD35
6F7C B9F56F     9020     LDA FDLISEQ,Y ; find a flag
6F7F C90A       9040     CMP #10     ; valid flag #?
6F81 B008       9060     BCS RSTD40  ; no - branch
6F83 AA         9080     TAX         ; else get this flag's onscreen Y
6F84 BDCE6D     9100     LDA FYL,X
6F87 C5DF       9120     CMP R+11    ; current flag displayed above this?
6F89 9012       9140     BCC RSTD48  ; no - branch
6F8B            9160 RSTD40
6F8B 84E1       9180     STY R+13    ; current flag goes in this slot
6F8D A004       9200     LDY #MAXFON ; max # flags on screen at once
6F8F            9220 RSTD44
6F8F B9F56F     9240     LDA FDLISEQ,Y
6F92 99F66F     9260     STA FDLISEQ+1,Y
6F95 C4E1       9280     CPY R+13
6F97 F008       9300     BEQ RSTD50
6F99 88         9320     DEY
6F9A 4C8F6F     9340     JMP RSTD44
6F9D            9360 RSTD48
6F9D C8         9380     INY
6F9E 4C7C6F     9400     JMP RSTD35
                9420 ;
6FA1            9440 RSTD50
6FA1 A5E0       9460     LDA R+12
6FA3 99F56F     9480     STA FDLISEQ,Y ; put flag in slot
6FA6 60         9500     RTS
                9520 ;
                9540 ;
                9560 ;
                9580 ; MISSILE DATA FOR FLAGS
                9600 ;
6FA7            9620 MFLGDAT
6FA7 00007078   9640     .BYTE 0,0,$70,$78,$7C,$7E,$7E,$7C,$78,$70,$40,$40,$40,
                                $40,$40,$40,0,0
6FAB 7C7E7E7C
6FAF 78704040
6FB3 40404040
6FB7 0000
                9660 ;
                9680 ;
                9700 ; FLAG DLI ROUTINE SETS COLOR
                9720 ; AND HORIZONTAL POSITION OF
                9740 ; FLAGS.
                9760 ; FLAGS IN "FDLISEQ" TABLE ARE
                9780 ; HANDLED IN ORDER AND INDEXED
```

```
                    9700 ; FLAG DLI ROUTINE SETS COLOR
                    9720 ; AND HORIZONTAL POSITION OF
                    9740 ; FLAGS.
                    9760 ; FLAGS IN "FDLISEQ" TABLE ARE
                    9780 ; HANDLED IN ORDER AND INDEXED
```



```
                    9800 ; WITH "FDLINDX"
                    9820 ;
   6FB9             9840 FDLI
   6FB9 48          9860     PHA
   6FBA 98          9880     TYA
   6FBB 48          9900     PHA
   6FBC 8A          9920     TXA
   6FBD 48          9940     PHA
                    9960 ;
   6FBE AEFA6F      9980     LDX FDLINDX ; get index to next flag
   6FC1 BDF56F      010000   LDA FDLISEQ,X ; get next flag #
   6FC4 AA          010020   TAX
   6FC5 BDE26D      010040   LDA FCOLOR,X ; set color
   6FC8 2901        010060   AND #1       ; use color or white?
   6FCA F005        010080   BEQ FDLI10   ; use color - branch
   6FCC A90F        010100   LDA #$0F     ; use white
   6FCE 4CD46F      010120   JMP FDLI20
   6FD1             010140 FDLI10
   6FD1 BDE26D      010160   LDA FCOLOR,X
   6FD4             010180 FDLI20
   6FD4 8D19D0      010200   STA $D019
                    010220 ;
   6FD7 BCEC6D      010240   LDY FHPOS,X ; set horiz pos.
   6FDA 8C07D0      010260   STY HPOSM3
   6FDD C8          010280   INY
   6FDE C8          010300   INY
   6FDF 8C06D0      010320   STY HPOSM2
   6FE2 C8          010340   INY
   6FE3 C8          010360   INY
   6FE4 8C05D0      010380   STY HPOSM1
   6FE7 C8          010400   INY
   6FE8 C8          010420   INY
   6FE9 8C04D0      010440   STY HPOSM0
                    010460 ;
                    010480 ;
   6FEC EEFA6F      010500   INC FDLINDX ; inc index for next DLI / next flag
                    010520 ;
   6FEF 68          010540   PLA
   6FF0 AA          010560   TAX
   6FF1 68          010580   PLA
   6FF2 A8          010600   TAY
   6FF3 68          010620   PLA
   6FF4 40          010640   RTI
                    010660 ;
                    010680 ;
                    010700 ; FDLISEQ table holds order of
                    010720 ; flags on screen from top to
                    010740 ; bottom for DLI's to access.
                    010760 ; Max number of flags on screen
                    010780 ; is 8.
                    010800 ;
      =0004         010820 MAXFON = 4     ; max # of flags on screen at one time
   6FF5             010840 FDLISEQ *= *+MAXFON+1
   6FFA             010860 FDLINDX *= *+1 ; index to FDLISEQ
                    010880 ;
                    010900 ;
                    010920 ;
   6FFB             010940 FTOPDLI *= *+1 ; flag for 1st DLI
```

Moore Business Forms, Inc. 141ZKH

```
6FFB                010920 ;
                    010940 FTOPDLI *= *+1 ; flag for 1st DLI
```



```
                    010960 ;
                    010980 ;
                    014620 ;
                    014640 ;
                    014660 ;;.OPT LIST
      =6FFC         014680 PROGEND = *
                    014700 ;;.OPT NO LIST
6FFC                014720    .END
```

ASSEMBLY ERRORS: 0   17093 BYTES FREE

Moore Business Forms, Inc. sv 1412RH

```
 5B17 ACCCNT      =0028 ACCDLY      5FEE ADDP1       53C2 ADDS10
 53C6 ADDS20      53AE ADDSCOR     =004D ATRACT     =D201 AUDC1
=D208 AUDCTL     =D200 AUDF1       676A AUDV1      =676C AUDV2
=676E AUDV3      =6770 AUDV4
 6B8F BAKSAV      5B0F BLSTINC     5B10 BLSTNZR     5B12 BPTHCT
 6701 BR419       6708 BR420       673C BR421       6747 BR423
 674F BR424       671B BR457       6726 BR458       672C BR459
 66FE BR461       66F4 BR463       6B87 BSTBLH      6B7F BSTBLL
 5B11 BUGDIR      6018 BUGH0       6019 BUGH1       601A BUGH2
 601B BUGH3       601C BUGP0       60DC BUGP1       619C BUGP2
 625C BUGP3       42A6 BUGX        42A7 BUGY       =FFF0 BWDSPD
 5C59 BYTADD
 4294 CARLFT     =02FC CH         =02F2 CH1         420A CHARH
 424A CHARL      =9000 CHARS      =02F4 CHBAS       55F5 CHCOPY
 52E3 CHGDSP      4012 CHINDX      4B8A CHRREF      658B COLI10
 659E COLI20      65B0 COLI30      65B4 COLI40      65B9 COLI50
 6589 COLISN     =D012 COLPM0     =D013 COLPM1     =D014 COLPM2
=D015 COLPM3     =D01F CONSOL      4013 CURCHAR     400E CX
 400F CY
=3740 DATAX       6916 DATE05      6920 DATE10      6931 DATE15
 6965 DATE20      69AB DATE30      69C4 DATE40      6912 DATEXP
 5A74 DESE10      5A84 DESE15      5A9E DESE20      5AEC DESE30
 5AC9 DESE56      5ACB DESE57      5A69 DESETUP     6338 DIR
 5C53 DIRINC      53FA DLI1        541E DLI2        547E DLI220
 42A8 DLIST       6870 DNSC01      687D DNSC02      68B5 DNSC10
 68C4 DNSC15      6844 DNSCRN      59BC DRA100      59B1 DRA110
 59A8 DRAW        59C3 DRAW01      59C7 DRAW02      59CE DRAW04
 59D4 DRAW05      59D6 DRAW10      59FC DRAW15      5A0A DRAW20
 5A0B DRAW30      6EEF DRWF05      6EF4 DRWF20      6F01 DRWF21
 6F06 DRWF22      6F18 DRWF30      6F2E DRWF40      6F3F DRWF50
 6EC8 DRWFLG      4008 DSCROL      411B DSPTBH      403C DSPTBL
 69E5 DXPTR1H     69E1 DXPTR1L     69ED DXPTR2H     69E9 DXPTR2L
 5480 ECAR        548A ECAR10      548E ECAR20      54A8 ECAR30
 54B0 ECAR40      54B7 ECAR50      54CB ECAR60      554A ECARDAT
 5501 EFLA10      5527 EFLA20      54FB EFLAG      =002D ENDJMP
 5A22 ERAS02      5A29 ERAS03      5A30 ERAS04      5A36 ERAS05
 5A29 ERAS06      5A38 ERAS10      5A58 ERAS15      5A66 ERAS30
 5A0E ERASE
 6DE2 FCOLOR      6FB9 FDLI        6FD1 FDLI10      6FD4 FDLI20
 6FFA FDLINDX     6FF5 FDLISEQ    =0050 FDLY        6D21 FDLYCT
 69C8 FFBYTE      6D20 FHIT        6DEC FHPOS       660B FLAG
 552A FLAGDAT     6D1F FLGCAR      6D1F FLGCARM     6D2A FLGH04
 6D37 FLGH06      6D3C FLGH10      6D51 FLGH20      6D65 FLGH30
 6D69 FLGH40      6D6F FLGH50      6D83 FLGH60      6DA8 FLGH62
 6DAB FLGH64      6DB3 FLGH70      6DBF FLGH74      6DC2 FLGH76
 6DC8 FLGH80      6DCD FLGH90      6D22 FLGHIT      6E5F FLGS01
 6E7A FLGS10      6E82 FLGS15      6E92 FLGS20      6E9A FLGS30
 6E9D FLGS40      6EA0 FLGS45      6EA1 FLGS50      6EA3 FLGS55
 6EB0 FLGS60      6EC1 FLGS70      6EC4 FLGS80      6E58 FLGSCR
 6D1F FTOGL       6FFB FTOPDLI    =0010 FWDSPD      6DD8 FYH
 6DCE FYL
=026F GPRIOR     =D01D GRACTL
 6E53 HFPOS      =D01E HITCLR      47B4 HMEM       =0370 HMEMSIZ
=D004 HPOSM0     =D005 HPOSM1     =D006 HPOSM2     =D007 HPOSM3
=D000 HPOSP0     =D001 HPOSP1     =D002 HPOSP2     =D003 HPOSP3
 5293 IDLI10      52A9 IDLI30      526B IDLIST      6E16 IHFL10
 6E2D IHFL20      6E38 IHFL30      6E14 IHFLGS      6E0A IHFPFC
```

Moore Business Forms, Inc. sv 1412RH

```
=D005 HPOSM1    =D006 HPOSM2    =D007 HPOSM3
=D000 HPOSP0    =D001 HPOSP1    =D002 HPOSP2    =D003 HPOSP3
 5293 IDLI10     52A9 IDLI30     526B IDLIST     6E16 IHFL10
 6E2D IHFL20     6E38 IHFL30     6E14 IHFLGS     6E0A IHFPFC

 6E00 IHFPFH     6DF6 IHFPFL     6E4E IHFPM      54CE IHME10
 54E5 IHME20     54F2 IHME30     54F4 IHME40     54CC IHMEM
 5562 IHTXT      5088 ILOOP      6423 IOBJ10     641F IOBJTB
=7800 IPF        50E1 IPFD10     50F2 IPFD20     50FB IPFD30
 50C7 IPFDATA    53DF IPLY10     53DB IPLYRDT    509C IPME10
 50A9 IPME20     5089 IPMEM      6772 ISFX       6776 ISFX10
 521D ISME10     5260 ISME30     520C ISMEM      530D ITAB05
 5315 ITAB10     532F ITAB15     5340 ITAB20     534F ITAB28
 5354 ITAB30     535E ITAB40     5379 ITAB50     537E ITAB60
 538B ITAB70     5301 ITABLES    55BC ITCH10     55D7 ITCH15
 55D9 ITCH20     55EF ITCH30     55B2 ITCHARS    53A7 IVAR10
 5394 IVARS      50FC IVBI
 5C74 JMPADD     5B14 JMPCTR    =0006 JMPDLY     5B18 JMPNDX
 5CA1 JMPPIC
 400C KX         400D KY
 6365 LASTINC    636E LASTNZR    42A5 LINCNT    =0004 LL
 66DF LP405     =0008 LR
=7300 M0         4B24 M1OTBL     5F76 M32TBL    =D00B M3PL
 4B3A M3TBL      41FA M4TBL      4B2A M5TBL      515B MAIN08
 5121 MAIN10     515E MAIN15     516D MAIN20     517C MAIN30
 5188 MAIN50    =2E80 MASKX     =0004 MAXFON     6FA7 MFLGDAT
 6353 MODE       52EC MOVD10     6380 MOVE       6382 MOVE10
 6387 MOVE15     638B MOVE16     6395 MOVE20     63B6 MOVE25
 63C5 MOVE30     63C6 MOVE40     6417 MOVE50
=0002 NACTYPE    65BB NEWRND    =D40E NMIEN     =0008 NPATH
=0006 NTYPES
 69F1 OBJDAT    =0001 OBJMAX     6A31 OBJMSK     69C7 OBJREF
 5B16 OLDBDIR    5B15 OLDBUGY    4011 OLDHPOS    4010 OLDVPOS
=4000 ORG       =E000 OSCHARS    642E OVRL05     6445 OVRL10
 6457 OVRL20     645B OVRL30     645C OVRL40     642A OVRLAP
 6377 OVRLAY
=7400 P0        =7500 P1        =7600 P2        =7700 P3
=D302 PACTL      635C PATH       5C5C PATHD      5AF9 PAUS10
 5B07 PAUS20     5B0E PAUS50     5AEE PAUSE     =02C0 PCOLR0
=02C1 PCOLR1    =02C2 PCOLR2    =02C3 PCOLR3     43B4 PFDATA
=026C PFLEN      68D4 PFLI10     68EC PFLI20     68C8 PFLINE
 4295 PLYRDAT   =D407 PMBASE    =7000 PMEM       5D44 PMO330
 5D54 PMO332     5D5E PMO333     5D65 PMO334     5D7E PMO342
 5E0A PMO625     5E20 PMO628     5CF2 PMOV10     5D14 PMOV20
 5D1C PMOV25     5D1F PMOV30     5D34 PMOV31     5D2C PMOV32
 5D3F PMOV33     5D79 PMOV34     5D86 PMOV35     5DB9 PMOV40
 5D9D PMOV42     5DB6 PMOV45     5DCD PMOV46     5DDC PMOV60
 5DE7 PMOV61     5DE4 PMOV611    5E34 PMOV62     5E44 PMOV63
 5E49 PMOV64     5E59 PMOV66     5E61 PMOV68     5E6B PMOV70
 5E70 PMOV75     5E83 PMOV80     5E90 PMOV90     5CCF PMOVE
=D300 PORTA      69D5 PPH        69C9 PPL       =6FFC PROGEND
 4000 PSCOR0     4001 PSCOR1     4002 PSCOR2
=00D4 R         =D20A RANDOM     5E91 RDSTAT1    5EBF RDSTAT2
 631C REDRWO     6AFF REFL10     6B07 REFL20     6B12 REFL30
 6B14 REFL40     6B16 REFL50     6AFD REFLECT    6B2F RFLPAT
 4003 ROUND      428A ROWBUF     6E6D RSTD08     6F57 RSTD10
 6F6A RSTD30     6F75 RSTD31     6F7A RSTD32     6F7C RSTD35
 6F8B RSTD40     6F8F RSTD44     6F9D RSTD48     6FA1 RSTD50
 6F42 RSTDLI    =0012 RTCLOK
 4014 SAVBUF     67DB SCR233     67D4 SCR242     67D8 SCR244
 4009 SCRCTR     5B13 SCRFLG     4004 SCRFLG2    67A2 SCRO08
 67B3 SCRO22     67C4 SCRO23     67C5 SCRO24     67ED SCRO25
 67F0 SCRO27     67FB SCRO30    =00CA SCROLC     400B SCROLF
```

```
=D300 PORTA     69D5 PPH     69C9 PPL     =8FFC PROGEND
 4000 PSCOR0     4001 PSCOR1     4002 PSCOR2
=00D4 R         =D20A RANDOM    5E91 RDSTAT1    5EBF RDSTAT2
 631C REDRWO     6AFF REFL10     6B07 REFL20     6B12 REFL30
 6B14 REFL40     6B16 REFL50     6AFD REFLECT    6B2F RFLPAT
 4003 ROUND      428A ROWBUF     6E6D RSTD08     6F57 RSTD10
 6F6A RSTD30     6F75 RSTD31     6F7A RSTD32     6F7C RSTD35
 6F8B RSTD40     6F8F RSTD44     6F9D RSTD48     6FA1 RSTD50
 6F42 RSTDLI    =0012 RTCLOK
 4014 SAVBUF     67DB SCR233     67D4 SCR242     67D8 SCR244
 4009 SCRCTR     5B13 SCRFLG     4004 SCRFLG2    67A2 SCRO08
 67B3 SCRO22     67C4 SCRO23     67C5 SCRO24     67ED SCRO25
 67F0 SCRO27     67FB SCRO30    =00CA SCROLC     400B SCROLF
```



```
 6793 SCROLL     400A SCRSPD    =0231 SDLSIH    =0230 SDLSTL
=022F SDMCTL     66DD SFX        6A7E SHAP05     6A82 SHAP10
 6A98 SHAP30     6AB6 SHAP40     6A71 SHAPEX     6AC1 SHIF10
 6AC3 SHIF15     6AC5 SHIF20     6AEE SHIF30     6ABD SHIFT
 5F14 SHOC03     5F1F SHOC05     5F3E SHOC07     5F47 SHOC08
 5F54 SHOC10     5F72 SHOC20     5F75 SHOC30     5EED SHOCAR
=D008 SIZEP0    =D009 SIZEP1    =D00A SIZEP2    =D00B SIZEP3
=D20F SKCTL     =7E70 SMEM      4005 SNDCTL     634A SPDCTR
 6341 SPEED     =0232 SSKCTL    663F SSTR1      661B SSTR2
 6663 SSTR3      6695 SSTR32     66C7 SSTR4      6D1F SWEEPR
 6D1F SWEEPRM
 5800 TCHARS     5465 TFDLI      6556 TIN100     64C2 TIN550
 64B8 TIN551     64BD TIN552     64B3 TIN553     6467 TINI10
 6471 TINI15     6476 TINI20     6478 TINI30     6481 TINI40
 64A1 TINI50     64AD TINI52     64C7 TINI54     64CF TINI56
 64E6 TINI60     64FB TINI62     6501 TINI64     650C TINI66
 6515 TINI70     6527 TINI74     652C TINI80     654B TINI90
 6462 TINIT      657C TMAXSP     6582 TMINSP     6588 TPNDX
 655E TPROB      5C4E TRA100     5B1E TRAC02     5B39 TRAC05
 5B41 TRAC06     5B4B TRAC10     5B70 TRAC18     5B74 TRAC20
 5B85 TRAC30     5B93 TRAC40     5BBC TRAC48     5B9C TRAC50
 5BCC TRAC56     5BD7 TRAC60     5BDF TRAC64     5BE1 TRAC65
 5BE1 TRAC70     5C18 TRAC80     5C25 TRAC90     5B1A TRACK
=D010 TRIGO      65DC TRKDIE     6BDF TRUCK      6C7F TRUCKM
 4006 TRUCKX     4007 TRUCKY     631D TYPE       69C5 TYPREF
=0001 UL         6825 UPSC10     682F UPSC15     683D UPSC18
 67FC UPSCR1     67FC UPSCRN    =0002 UR
 5190 VBINO5     51FF VBIN32     5201 VBIN40     518E VBINT
=D40B VCOUNT     6752 VOICE1    =6754 VOICE2    =6756 VOICE3
=6758 VOICE4     675A VPTR1     =D405 VSCROL     6762 VTIMR1
=0224 VVBLKD     65EC VWDIE      65CC VWJUMP
 5F9E WHEE05     5FA6 WHEE10     5FE1 WHEE20     5F7E WHEELS
 5FE2 WHLPOS     5B19 WHLSEQ    =D40A WSYNC
=000F XCRSH     =E462 XITVBV     6326 XPOS
=000E YCRSH      632F YPOS      =000B YTRKPT
```

Moore Business Forms, Inc. sv 1412RH